



TUGAS AKHIR – TI 141501

OPTIMASI MULTIPLE CAN-ORDER LEVEL PADA CAN-ORDER POLICY MENGGUNAKAN ALGORITMA SIMULATED ANNEALING (STUDI KASUS: SPARE PART INVENTORY PT INDONESIA ASAHAN ALUMINIUM (PERSERO))

FALDY MAULANA YUANTORO
NRP. 02411440000047

Dosen Pembimbing
Prof. Ir. Budi Santosa, M.Sc., Ph.D.
NIP. 196905121994021001

Departemen Teknik Industri
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya
2018



FINAL PROJECT – TI 141501

MULTIPLE CAN-ORDER LEVEL FOR CAN-ORDER POLICY OPTIMIZATION USING SIMULATED ANNEALING ALGORITHM (CASE STUDY: SPARE PART INVENTORY PT INDONESIA ASAHAN ALUMINIUM (PERSERO))

FALDY MAULANA YUANTORO
NRP. 02411440000047

Supervisor
Prof. Ir. Budi Santosa, M.Sc., Ph.D.
NIP. 196905121994021001

Department of Industrial Engineering
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2018

LEMBAR PENGESAHAN

**OPTIMASI *MULTIPLE CAN-ORDER LEVEL* PADA *CAN-ORDER POLICY*
MENGUNAKAN ALGORITMA *SIMULATED ANNEALING* (STUDI
KASUS: *SPARE PART INVENTORY* PT INDONESIA ASAHAN
ALUMINIUM (PERSERO))**

TUGAS AKHIR


Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik
Program Studi S-1 Departemen Teknik Industri
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Oleh:

FALDY MAULANA YUANTORO
NRP. 02411440000047

Disetujui oleh,

Dosen Pembimbing Tugas Akhir


Prof. Ir. Budi Santosa, M.Sc., Ph.D.
NIP. 196905121994021001

SURABAYA, JANUARI 2018

**OPTIMASI *MULTIPLE CAN-ORDER LEVEL* PADA *CAN-ORDER POLICY*
MENGUNAKAN ALGORITMA *SIMULATED ANNEALING* (STUDI
KASUS: *SPARE PART INVENTORY* PT INDONESIA ASAHAN
ALUMINIUM (PERSERO))**

Nama : Faldy Maulana Yuantoro
NRP : 02411440000047
Pembimbing : Prof. Ir. Budi Santosa, M.Sc., Ph.D.

ABSTRAK

Persediaan merupakan stok dari barang-barang yang disimpan dan akan digunakan di masa yang akan datang. Salah satu permasalahan yang terjadi adalah apabila permintaan pada suatu *item* tergolong permintaan yang *intermittent*. Kebijakan persediaan pada jenis permintaan tersebut sulit untuk ditentukan karena pola permintaan yang tidak pasti sehingga tingkat kesalahan prediksinya besar. Permasalahan jenis permintaan tersebut muncul pada kebutuhan *spare part* di PT Indonesia Asahan Aluminium (Persero). Untuk mengatasinya, model kebijakan *can order policy* menggunakan *multiple can-order level* (s_i , c_{ij} , S_{ij}) dapat menyelesaikan permasalahan dengan *joint replenishment* untuk mengkoordinasikan pesanan antar item. Model tersebut diketahui merupakan *integer non-linear programming* (INLP) yang dapat digolongkan ke dalam permasalahan *NP-Hard*. Pendekatan metaheuristik yaitu *simulated annealing* akan digunakan untuk mendapatkan solusi yang memuaskan dengan waktu penyelesaian yang cepat. Metode *Global Criterion* juga akan digunakan untuk mendapatkan fungsi *multi objective*, yaitu meminimasi total biaya persediaan dan meminimasi jumlah *carrier supplier*. Berdasarkan eksperimen dengan 3 *supplier* dan 157 *item* didapatkan hasil yang lebih baik dari kebijakan persediaan di PT INALUM (Persero) dengan adanya penghematan dari kondisi eksisting yang diterapkan perusahaan sebesar 2.6% pada jumlah *carrier* dan penghematan biaya sebesar 14.43% atau sebesar \$81,570.79 (Rp 1,060,420,270).

Kata Kunci : *multiple can-order level, can order policy, spare part, simulated annealing, metaheuristik*

(Halaman Ini Sengaja Dikosongkan)

**MULTIPLE CAN-ORDER LEVEL FOR CAN-ORDER POLICY
OPTIMIZATION USING SIMULATED ANNEALING ALGORITHM (CASE
STUDY: SPARE PART INVENTORY PT INDONESIA ASAHAN
ALUMINIUM (PERSERO))**

Name : Faldy Maulana Yuantoro
NRP : 02411440000047
Supervisor : Prof. Ir. Budi Santosa, M.Sc., Ph.D.

ABSTRACT

Inventory is the stock of goods that is stored and will be used for the future needs. One of the problems that occurs is when the demand on an item belongs to an intermittent request. Inventory policy on this type of demand is difficult to be determined due to an uncertain demand pattern, so that the predictive error rate is increased. This type of demand problem arises on the needs of spare parts at PT Indonesia Asahan Aluminium (Persero). To overcome this problem, the can order policy model using multiple can-order level (s_i , c_{ij} , S_{ij}) is able to solve joint replenishment problem to coordinate orders between items. The can order policy model built is in the form of integer non-linear programming (INLP) that can be classified into NP-Hard problems. The metaheuristic approach, Simulated Annealing, will be used to obtain a satisfied solution with a fast computation time to solve the model built. The Global Criterion method will also be used to obtain multi-objective functions, which can minimize total inventory cost and supplier carrier quantity. Based on experiments with 3 suppliers and 157 items, a better result of the inventory policy at PT INALUM (Persero) is obtained with savings from existing conditions applied by the company of 2.6% on carrier amount and cost savings of 14.43% or \$ 81,570.79 (Rp 1,060,420,270).

Keywords : *multiple can-order level, can order policy, spare part, simulated annealing, metaheuristics*

(Halaman Ini Sengaja Dikosongkan)

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Allah SWT. yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Optimasi *Multiple Can-Order Level* pada *Can-Order Policy* Menggunakan Algoritma *Simulated Annealing* (Studi Kasus: *Spare Part Inventory* PT Indonesia Asahan Aluminium (Persero))”**

Penyusunan Tugas Akhir ini dilakukan untuk memenuhi persyaratan menyelesaikan studi Strata-1 di Departemen Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Dalam penyusunan Tugas Akhir ini tidak terlepas dari bantuan banyak pihak yang telah memberikan bantuan kepada penulis. Oleh karena itu penulis mengucapkan terimakasih kepada:

1. Prof. Ir. Budi Santosa, M.S., Ph.D. selaku selaku dosen pembimbing yang telah memberikan pembelajaran, arahan, dan masukkan hingga selesainya Tugas Akhir.
2. Ibu Neng Ratnawaty selaku staf Seksi Smelter Corporate Development (SCD) dan Bapak Pra F. Alfredo selaku karyawan di seksi *Spare Parts Warehouse* (SWH) PT INALUM (Persero) yang telah bersedia untuk membimbing dan membantu penulis mencari data yang dibutuhkan selama pengerjaan tugas akhir.
3. Ir. Hari Supriyanto, MSIE. selaku dosen wali atas semua bimbingannya selama ini.
4. Bapak Nurhadi Siswanto, S.T., MSIE., Ph.D. selaku Kepala Departemen Teknik Industri ITS atas kelancaran birokrasi selama pengerjaan Tugas Akhir.
5. Ibu Effi Latiffianti, S.T, M.T. dan Bapak Dody Hartanto, S.T., M.T. selaku dosen penguji sidang tugas akhir dan segenap dosen dan karyawan Departemen Teknik Industri ITS yang telah membimbing penulis selama menempuh masa studi.
6. Ibu Sulisty Perjuaningsih, Bapak Yani Abriyantoro, Faizha Adhania S. Y., dan Farah Nurfadhilla Y. selaku kedua orang tua dan saudara kandung penulis

yang senantiasa memberikan dukungan, inspirasi dan kepercayaan yang luar biasa besar kepada penulis.

7. Asisten Laboratorium QMIPA, Azaria Zada, dan kerabat-kerabat penulis lainnya yang selalu memberikan inspirasi dan mendorong penulis untuk menyelesaikan tugas akhir.

Laporan Tugas Akhir ini tidak luput dari kekurangan. Jika terdapat kesalahan dalam penulisan laporan, penulis mengucapkan permohonan maaf. Segala kritik dan saran sangat penulis harapkan sebagai masukan bagi penulis. Penulis berharap agar Laporan Tugas Akhir ini bermanfaat bagi semua yang membutuhkan.

Surabaya, Januari 2018

Penulis

DAFTAR ISI

ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	5
1.5 Ruang Lingkup Penelitian.....	5
1.5.1 Batasan	5
1.5.2 Asumsi.....	6
1.6 Sistematika Penulisan.....	6
BAB 2 TINJAUAN PUSTAKA	9
2.1 Manajemen <i>Inventory</i>	9
2.1.1 Biaya Persediaan	10
2.1.2 Klasifikasi Demand.....	11
2.2 <i>Inventory Policy</i>	12
2.2.1 Joint Replenishment Policy.....	14
2.3 <i>Can-Order Policy</i>	15
2.3.1 Multiple Can-order Level pada Can-order Policy.....	15
2.4 Algoritma <i>Simulated Annealing</i> untuk Optimasi	20
2.5 Optimasi <i>Multi-Objective</i>	22
2.5.1 Metode Global Criterion	22
2.6 Posisi Penelitian	23
BAB 3 METODOLOGI PENELITIAN.....	27
3.1 Deskripsi Permasalahan	28
3.2 Studi Literatur	30

3.3	Pengumpulan Data.....	30
3.4	Tahap Pengembangan Model Matematis	31
3.5	Tahap Pengembangan Algoritma <i>Simulated Annealing</i>	34
3.6	Validasi Model dan Algoritma	39
3.7	Eksperimen	40
3.8	Analisis dan Interpretasi Hasil.....	40
3.9	Penarikan Kesimpulan dan Saran	40
BAB 4	PENGEMBANGAN MODEL DAN ALGORITMA.....	41
4.1	Pengembangan Algoritma <i>Simulated Annealing</i>	41
4.2	Validasi dan Verifikasi	49
4.2.1	Validasi dan Verifikasi Model Matematis	49
4.2.2	Validasi dan Verifikasi Algoritma.....	53
BAB 5	EKSPERIMEN DAN ANALISIS	61
5.1	Pengumpulan Data.....	61
5.2	Kondisi Eksisting.....	64
5.3	Eksperimen Uji Parameter Algoritma SA	65
5.3.1	Deskripsi Data Uji	66
5.3.2	Eksperimen Data Uji 1	66
5.3.3	Eksperimen Data Uji 2	68
5.3.4	Eksperimen Data Uji 3	71
5.4	Eksperimen pada Studi Kasus PT INALUM (Persero)	75
5.5	Analisis Hasil Eksperimen.....	76
5.5.1	Analisis Kondisi Eksisting.....	77
5.5.2	Analisis Eksperimen Uji Parameter Algoritma SA	77
5.5.3	Analisis Hasil Penyelesaian Studi Kasus.....	78
5.5.4	Perbandingan Hasil Algoritma SA dengan Kondisi Eksisting	79
BAB 6	KESIMPULAN DAN SARAN	83
6.1	Kesimpulan.....	83
6.2	Saran	84
DAFTAR	PUSTAKA.....	85
LAMPIRAN A	87
LAMPIRAN B	89

LAMPIRAN C	101
LAMPIRAN D.....	105
LAMPIRAN E	117
LAMPIRAN F.....	125
BIOGRAFI PENULIS	149

(Halaman Ini Sengaja Dikosongkan)

DAFTAR GAMBAR

Gambar 1.1 Klasifikasi Tipe Permintaan Spare Part PT INALUM.....	2
Gambar 2.1 <i>Trade Off</i> Komponen Biaya <i>Inventory</i>	11
Gambar 2.2 Matriks Pembagian 4 Tipe <i>Demand</i>	12
Gambar 2.3 Konvensional <i>Can-Order Policy</i> pada 3 <i>item</i>	15
Gambar 2.4 <i>Multiple Can-order Level</i> pada Sistem (s, c, S) untuk 3 <i>item</i> yang ...	16
Gambar 3.1 <i>Flowchart</i> Metodologi Penelitian.....	27
Gambar 3.2 <i>Scatter Plot</i> Permintaan <i>Spare Part</i> PT INALUM.....	29
Gambar 3.3 Model Sistem <i>Multiple Can-Order Level</i> pada <i>Can-Order Policy</i>	31
Gambar 4.1 Skema Persediaan Hasil Pembangkitan Bilangan <i>Random</i> pada <i>Item</i> 1	45
Gambar 4.2 Skema Persediaan Hasil Pembangkitan Bilangan <i>Random</i> pada <i>Item</i> 2	45
Gambar 4.3 Skema Persediaan Hasil Pembangkitan Bilangan <i>Random</i> pada <i>Item</i> 3	46
Gambar 4.4 Hasil Perhitungan Fungsi Tujuan 1 dengan Metode Eksak pada Data Verifikasi.....	50
Gambar 4.5 Hasil Perhitungan Fungsi Tujuan 2 dengan Metode Eksak pada Data Verifikasi.....	51
Gambar 4.6 Hasil Perhitungan Fungsi Multi Tujuan dengan Metode Eksak pada Data Verifikasi	52
Gambar 4.7 Skema Persediaan Hasil Verifikasi Algoritma SA pada <i>Item</i> 1	56
Gambar 4.8 Skema Persediaan Hasil Verifikasi Algoritma SA pada <i>Item</i> 2.....	57
Gambar 4.9 Skema Persediaan Hasil Verifikasi Algoritma SA pada <i>Item</i> 3	57
Gambar 4.10 Plot Grafik Nilai Fungsi Multi Tujuan pada Algoritma SA.....	58
Gambar 5.1 Diagram Pareto <i>Supplier</i> pada <i>Spare Part</i>	62
Gambar 5.2 Hasil <i>Running Lingo</i> pada Data Uji 2	69
Gambar 5.3 Hasil <i>Running Lingo</i> pada Data Uji 3 Fungsi Tujuan 1.....	72
Gambar 5.4 Hasil <i>Running Lingo</i> pada Data Uji 3 Fungsi Tujuan 2.....	72

(Halaman Ini Sengaja Dikosongkan)

DAFTAR TABEL

Tabel 2.1 <i>Replenishment Policy</i> pada <i>Inventory Control</i>	13
Tabel 2.2 Posisi Penelitian	25
Tabel 3.1 Jumlah <i>Item</i> yang Dipenuhi Vendor 1 dan 2 pada Tahun 2018.....	29
Tabel 3.2 Gambaran Struktur Solusi <i>Re-order Level</i> (s).....	37
Tabel 3.3 Gambaran Struktur Solusi <i>Can-order Level</i> (c).....	37
Tabel 3.4 Gambaran Stuktur Solusi <i>Order-up-to Level</i> (S)	38
Tabel 4.1 Data Permintaan, Volume, dan Persediaan Awal untuk Model Validasi	42
Tabel 4.2 Data Biaya Persediaan untuk Model Validasi.....	42
Tabel 4.3 Nilai <i>re-order level</i> (s_i) yang Dibangkitkan secara Random	42
Tabel 4.4 Nilai <i>can-order level</i> (c_{ij}) yang Dibangkitkan secara <i>Random</i>	43
Tabel 4.5 Nilai <i>order-up-to level</i> (S_{ij}) yang Dibangkitkan secara <i>Random</i>	43
Tabel 4.6 Hasil Solusi Variabel biner q	44
Tabel 4.7 <i>Level</i> Persediaan Hasil Pembangkitan Variabel <i>Random</i>	44
Tabel 4.8 Jumlah Pemesanan Hasil Pembangkitan Variabel <i>Random</i>	45
Tabel 4.9 <i>Re-order Level</i> (s) Hasil Perhitungan Metode Eksak.....	52
Tabel 4.10 <i>Can-order Level</i> (c) Hasil Perhitungan Metode Eksak.....	52
Tabel 4.11 <i>Order-Up-To Level</i> (S) Hasil Perhitungan Metode Eksak.....	53
Tabel 4.12 Jumlah Pesanan Hasil Perhitungan Metode Eksak	53
Tabel 4.13 <i>Level</i> Persediaan Hasil Perhitungan Metode Eksak.....	53
Tabel 4.14 Jumlah <i>Item Shortage</i> Hasil Perhitungan Metode Eksak.....	53
Tabel 4.15 Perbandingan Hasil Verifikasi Algoritma SA dengan Perhitungan Eksak	58
Tabel 4.16 Perbandingan Hasil Verifikasi Algoritma SA dengan Perhitungan Eksak <i>Multi Supplier</i>	60
Tabel 5.1 Data <i>Supplier</i> dan Jumlah <i>Item</i> yang Disuplai.....	61
Tabel 5.2 Data Peramalan Permintaan <i>Spare Part</i> Tahun 2018	63
Tabel 5.3 Data Atribut <i>Spare Part</i>	63
Tabel 5.4 Biaya Pemesanan <i>Major Spare Part</i>	64
Tabel 5.5 Hasil Perhitungan Biaya Persediaan Kondisi Eksisting.....	65

Tabel 5.6 Hasil Perhitungan Total <i>Carrier</i> Kondisi Eksisting	65
Tabel 5.7 Jumlah Pesanan <i>Item</i> pada Kondisi Eksisting	65
Tabel 5.8 Data Uji	66
Tabel 5.9 Hasil Uji Parameter Data Uji 1	67
Tabel 5.11 Hasil Perhitungan Metode Eksak Data Uji 1	68
Tabel 5.12 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 1	68
Tabel 5.13 Hasil Uji Parameter Data Uji 2	69
Tabel 5.14 Hasil Uji Parameter Data Uji 2 (Lanjutan)	70
Tabel 5.15 Hasil Perhitungan Metode Eksak Data Uji 2	70
Tabel 5.16 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 2	71
Tabel 5.17 Hasil Uji Parameter Data Uji 3	73
Tabel 5.18 Hasil Perhitungan Metode Eksak Data Uji 3	74
Tabel 5.19 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 3	74
Tabel 5.20 Hasil <i>Running Model</i> pada Penyelesaian Studi Kasus	75
Tabel 5.21 Jumlah <i>Carrier</i> Hasil Algoritma SA	75
Tabel 5.22 Jumlah Item yang Dipesan dari Hasil Algoritma	76
Tabel 5.23 Hasil Biaya Algoritma SA	76
Tabel 5.24 Perbandingan Hasil Kondisi Eksisting dan Hasil Optimasi	79
Tabel 5.25 Perbandingan Jumlah <i>Carrier</i> Kondisi Eksisting dan Hasil Optimasi	79
Tabel 5.26 Perbandingan Biaya Kondisi Eksisting dan Hasil Optimasi	80

BAB 1

PENDAHULUAN

Pada bab pendahuluan ini berisi tentang hal-hal yang mendasari dilakukannya penelitian, serta pengidentifikasian masalah penelitian, yang meliputi latar belakang masalah, perumusan masalah, ruang lingkup penelitian, tujuan penelitian, dan manfaat penelitian.

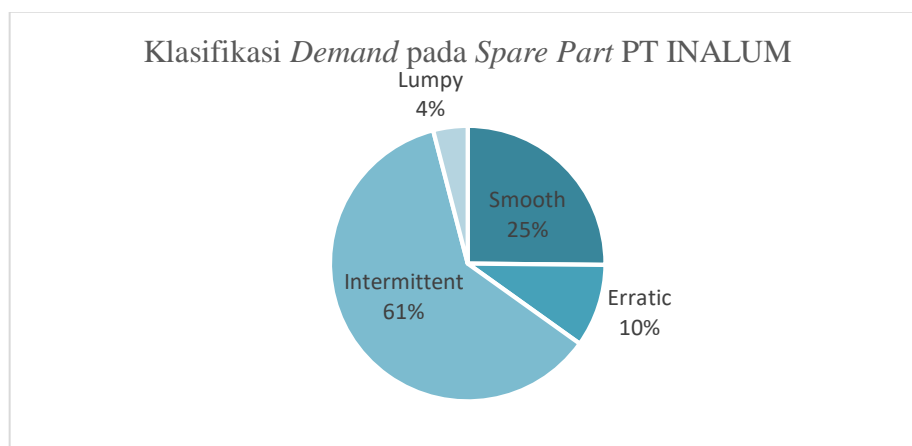
1.1 Latar Belakang

Inventory merupakan komoditas di dalam pengelolaan dari sebuah perusahaan yang digunakan pada periode tertentu untuk memenuhi permintaan di waktu yang akan datang (Sipper & Bulfin, 1997). Pada sektor manufaktur, komoditas yang dimaksud dapat berupa bahan baku, produk setengah jadi, produk jadi, maupun *spare part*. Perusahaan yang menginvestasikan persediaan, digunakan untuk mengantisipasi adanya perubahan pada permintaan dan untuk memproduksi barang pada jarak tertentu dari aktual *customer* (Vollmann, et al., 1997). Pengendalian *inventory* yang dilakukan secara efektif dapat berperan sangat penting dalam hubungannya dengan *supply chain management*. Pengendalian *inventory* dilakukan dengan menentukan apa yang dipesan, berapa jumlah pesanan, dan kapan pemesanan dilakukan. Umumnya, pengendalian *inventory* digunakan untuk mengurangi adanya investasi, seperti biaya simpan, biaya pemesanan, dan biaya pembelian, dan juga untuk meningkatkan *customer service level* dengan menghindari adanya *shortage*.

Strategi penentuan pengendalian *inventory* ini salah satunya didasarkan pada klasifikasi tipe permintaan yang berbeda. Permintaan dibagi berdasarkan 4 kategori, antara lain *smooth*, *erratic*, *intermittent*, dan *lumpy* (Syntetos AA, 2005). Kategori tersebut dibagi berdasarkan 2 parameter yaitu *average inter-demand interval* dan *squared coefficient of variation*. Permintaan *intermittent* dan *lumpy* sendiri sangat sulit untuk diprediksi dan tingkat *error*-nya sangat berpengaruh terhadap ketidakpastian jumlah stok di gudang dan permintaan. Variabilitas tidak hanya terjadi pada ukuran permintaan tapi juga pola permintaan itu sendiri (Ramaekers & Janssens, 2014). Permasalahan yang terjadi pada jenis permintaan

yang *intermittent* adalah sulitnya dalam menentukan strategi kebijakan persediaan karena pola permintaan yang tidak pasti dan sulit untuk diprediksi.

Permasalahan jenis permintaan tersebut juga muncul pada kebutuhan *spare part* di PT Indonesia Asahan Aluminium (Persero). PT Indonesia Asahan Aluminium atau yang biasa disebut PT INALUM (Persero) merupakan salah satu pabrik peleburan aluminium terbesar di Asia Tenggara. Untuk melakukan proses produksi dari bahan baku sampai produk jadi, perusahaan membutuhkan kinerja dari mesin-mesin produksi. Sehingga, keandalan dari mesin maupun *supply tools* dalam proses produksi merupakan salah satu faktor penting agar *demand* dari produk jadi tetap bisa dipenuhi seutuhnya. Untuk menjaga keandalan dari peralatan produksi tersebut, perusahaan melakukan strategi *maintenance*. Untuk melakukan *preventive maintenance*, dibutuhkan adanya *spare part*, yang mana *spare part* tersebut merupakan komponen-komponen pendukung dari mesin-mesin produksi. Saat *spare part* tersebut tidak tersedia, proses *maintenance* yang seharusnya dilakukan, menjadi tidak dapat dilakukan sampai *spare part* dari mesin tersebut tersedia. Hal ini dapat mengakibatkan berhentinya proses produksi, yang mengakibatkan *demand* dari *customer* tidak bisa terpenuhi. Hal tersebut menunjukkan bahwa pentingnya pengendalian *inventory* pada *spare part*. PT INALUM (Persero) mengendalikan *spare part inventory* sebanyak kurang lebih 14.000 jenis *item* yang terdiri dari *mechanical part*, *electrical part*, fabrikasi lokal, material mekanik, *general goods*, *tools* dan pelumas. Sebagian besar dari *part - part* tersebut dapat digolongkan ke dalam tipe *demand intermittent* seperti yang ditunjukkan pada gambar berikut.



Gambar 1.1 Klasifikasi Tipe Permintaan Spare Part PT INALUM
(Sumber: Data *Forecast Kebutuhan Spare Part Inventory* Tahun 2018)

Dari *mechanical part* yang dibutuhkan pada 2018, 61% diantaranya tergolong ke dalam permintaan *intermittent*. Prediksi permintaan *spare part* yang *intermittent* tersebut yang selalu sulit untuk ditentukan berapa jumlah *part* di dalam proses *maintenance*. Selain itu, pemesanan *spare part* yang dilakukan secara independen pada PT INALUM (Persero) menyebabkan tingginya *ordering cost*. Padahal, proses tender pada *spare part* tidak terlalu signifikan berpengaruh, karena harus melakukan pembelian *item* yang banyak jenisnya dan penawaran dilihat berdasarkan kesesuaian spesifikasi *part*. Sehingga, perlu dilakukan kebijakan *inventory* yang efektif agar biaya *inventory* tetap rendah, dan jumlah *item* yang *stockout* juga rendah.

Menurut Sipper & Bulfin (1997), *periodic review system* sangat direkomendasikan pada penerapan pengendalian *inventory* untuk *intermittent demand*, yaitu dengan menerapkan EOQ, sistem (S, T), sistem (s, S), maupun *replenishment* lainnya. Seperti yang dikemukakan Ghorbel et al. (2014), untuk mengatasi adanya variasi pada suplai kuantitas dan periode yang bervariasi, maka digunakan *replenishment policies* jenis (s, S). Pada *multi-item inventory control*, *joint replenishment problem* (JRP) mengoordinasikan antara jumlah kuantitas pesanan pada sejumlah *item* dalam satu gudang (Nagasawa & Irohara, 2016). Ketika ada beberapa *item* yang dipesan pada *supplier* yang sama, biaya pemesanan akan bergantung pada jumlah pesanan dan jumlah *item* yang dipesan. Walaupun begitu, jumlah pesanan dari *item* yang berbeda adalah independen walaupun memiliki *supplier* yang sama. Untuk mengatasinya, *Can-order Policy*, yang merupakan pengembangan dari kebijakan (s, S), dapat mengendalikan koordinasi pesanan yang mengakomodasi keterkaitan antar *item* (Putra & Pujawan, 2011) dan memiliki hasil yang lebih baik dari *uncoordinated policy* dengan menghasilkan 20% *saving* (Balintfy, 1964). *Can-order policy* dilakukan dengan menentukan 3 parameter (s, c, S), yaitu *re-order level* (s), *can-order level* (c), dan *order-up-to level* (S).

Pada formulasi model yang dibangun oleh Nagasawa & Irohara (2016), model kebijakan (s, c, S) diketahui merupakan *integer non-linear programming* (INLP). Sehingga, permasalahan seperti ini digolongkan pada permasalahan *NP-Hard*. Sehingga, jika diselesaikan dengan metode eksak, akan membutuhkan waktu

penyelesaian yang sangat lama karena *item* yang ditangani pada *spare part inventory* mencapai ratusan sampai ribuan *item*. Pendekatan metaheuristik akan digunakan untuk mendapatkan solusi yang memuaskan dengan waktu penyelesaian yang cepat. Nagasawa & Irohara (2016) menyelesaikan permasalahan tersebut menggunakan *Genetic Algorithm*. Jika GA digunakan pada *spare part inventory* yang memiliki ribuan jenis *item*, waktu penyelesaian setiap iterasi dapat menjadi lama karena solusi dibangun berdasarkan populasi.

Oleh karena itu, penulis membangun model *multi objective multiple can order policy* menggunakan *Simulated Annealing*, agar diharapkan waktu penyelesaiannya lebih cepat dengan hasil solusi yang tetap memuaskan karena SA merupakan Algoritma *neighborhood-based*. Pada masalah dengan ukuran data yang besar, SA lebih cepat menemukan solusi optimal dibandingkan GA. Algoritma SA juga lebih sederhana dengan input parameter yang lebih sedikit dari pada metode lain. Metode *Global Criterion* juga akan digunakan untuk mendapatkan fungsi *multi objective* yang mempertimbangkan fungsi tujuan berdasarkan selisihnya dengan solusi optimal. Fungsi tujuan dari model yang dikembangkan adalah untuk meminimasi total biaya persediaan dan meminimasi jumlah *carrier supplier* dengan memperhatikan kapasitas penyimpanan dan *multiple can-order level* untuk mengkoordinasikan *item* yang memiliki *supplier* yang sama.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, permasalahan pada penelitian ini adalah bagaimana menyelesaikan *multiple can-order level* pada *can-order policy joint replenishment problem* menggunakan algoritma *simulated annealing* pada studi kasus di *spare part inventory* PT Indonesia Asahan Aluminium (Persero)

1.3 Tujuan Penelitian

Beberapa tujuan yang dicapai dalam penelitian ini adalah sebagai berikut :

1. Memberikan rekomendasi dalam permasalahan *spare part inventory* PT Indonesia Asahan Aluminium (Persero) dengan menentukan jadwal

pemesanan dan kuantitas pesanan yang optimal dari setiap *item* untuk meminimasi total biaya dan jumlah angkutan *supplier*.

2. Mengembangkan algoritma *Simulated Annealing* untuk menyelesaikan permasalahan *multiple can-order level* pada *can-order policy*.

1.4 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah untuk memberikan kontribusi di bidang optimasi dengan penerapan algoritma *simulated annealing* untuk menyelesaikan permasalahan *multiple can-order level* pada *can-order policy* dan untuk memberikan alternatif solusi pada kebijakan *inventory control* di PT Indonesia Asahan Aluminium (Persero) untuk meminimasi total biaya persediaan dan total angkutan *supplier*.

1.5 Ruang Lingkup Penelitian

Dalam ruang lingkup penelitian akan dijelaskan mengenai batasan-batasan dan asumsi yang akan digunakan. Batasan digunakan agar penelitian tidak meluas dan tetap dalam lingkup penelitian, sedangkan asumsi digunakan untuk memberikan pembenaran pada kondisi penelitian terhadap realitasnya.

1.5.1 Batasan

Batasan yang digunakan pada penelitian *simulated annealing* untuk penyelesaian permasalahan *multiple can-order level* pada *can-order policy* adalah:

1. Permasalahan yang diteliti adalah *multiple can-order level* pada *can-order policy* dengan fungsi tujuan meminimasi total biaya persediaan dan meminimasi total *carrier*.
2. Data uji menggunakan data *spare part inventory* PT Indonesia Asahan Aluminium (Persero) pada tahun 2018.
3. *Item* yang diperhitungkan adalah *item* jenis *mechanical part* yang tergolong ke dalam *replacement part* dan *consumable part*.
4. *Supplier* yang diperhitungkan untuk dilakukan *joint replenishment* adalah 3 *supplier* yang memenuhi jumlah jenis *item* terbanyak pada tahun 2018.

5. Komputasi model dilakukan dengan jenis spesifikasi computer yang telah ditentukan sebelumnya.

1.5.2 Asumsi

Dan berikut ini adalah asumsi yang digunakan:

1. Seluruh input data bersifat deterministik selama periode perencanaan.
2. *Replenishment item* diantarkan dengan kapasitas jumlah *carrier* oleh *supplier* dan kapasitas Gudang yang tak terbatas.
3. Korelasi antar *item* yang dapat dilakukan *joint replenishment* dilihat berdasarkan kesamaan antar *supplier*.
4. Penalty *item* yang *shortage* didasarkan pada klasifikasi kritis ABC, *item* A memiliki *penalty* 50% dari biaya persediaan, *item* B 30%, dan *item* C 15%.
5. Kapasitas *carrier* setiap *supplier* ditentukan perusahaan sebesar 1.045.440 cm³.

1.6 Sistematika Penulisan

Laporan Tugas Akhir ini akan terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan ini berisi tentang hal-hal yang mendasari dilakukannya penelitian serta pengidentifikasian masalah penelitian. Bagian-bagian yang terdapat dalam bab pendahuluan ini meliputi latar belakang masalah, perumusan masalah, ruang lingkup penelitian, tujuan penelitian, serta manfaat penelitian.

BAB II TINJAUAN PUSTAKA

Tinjauan pustaka menguraikan teori, temuan, dan bahan penelitian lain yang diperoleh dari acuan yang akan dijadikan landasan untuk melakukan kegiatan penelitian yang akan dijadikan tugas akhir.

BAB III METODOLOGI PENELITIAN

Bab ini menguraikan metodologi penelitian yang dilakukan dalam mengembangkan algoritma *simulated annealing* untuk menyelesaikan permasalahan *multiple can-order level* pada *can-order policy*.

BAB IV PENGEMBANGAN MODEL DAN ALGORITMA

Pada bab pengembangan model dan algoritma ini akan dijelaskan tentang tahap-tahap yang dilakukan dalam mengembangkan model *multi objective multiple can-order level* dan algoritma *simulated annealing* yang digunakan dalam penyelesaian permasalahan *joint replenishment can-order policy*.

BAB V EKSPERIMEN DAN ANALISIS

Pada bab eksperimen dan analisis akan dijabarkan mengenai analisis hasil eksperimen penyelesaian studi kasus dari bab pengujian algoritma.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan hasil penelitian dan saran-saran yang berkaitan dengan penelitian selanjutnya.

(Halaman Ini Sengaja Dikosongkan)

BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan diuraikan mengenai teori, temuan dan hasil penelitian terdahulu dalam menyelesaikan permasalahan *multiple can-order level* pada *can-order policy*.

2.1 Manajemen *Inventory*

Setiap perusahaan selalu memiliki aset yang disimpan untuk keperluan tertentu. Salah satu aset yang disimpan dapat berupa persediaan atau *inventory*. *Inventory* merupakan stok dari barang-barang yang disimpan dan akan digunakan di masa yang akan datang untuk keperluan penjualan maupun keperluan lainnya (Fogarty, et al., 1991). Komoditas dari persediaan dapat diklasifikasikan menjadi 5 tipe (Vrat, 2014), antara lain:

- Persediaan *raw material* sebagai input untuk sistem manufaktur.
- Persediaan *bought-out-parts* (BOP) yang digunakan langsung menuju *assembly* dari produk.
- *Work-in-process* (WIP) *inventory* atau *pipeline inventory*.
- *Finished good inventory* yang akan didistribusikan ke *customer*.
- *Maintenance, repair, dan operating* (MRO) *supply*, termasuk *spare part*, *indirect material*, dan peralatan tambahan lain untuk sistem produksi dan servis.

Tujuan, kebijakan, dan keputusan dalam manajemen *inventory* haruslah konsisten dengan keseluruhan tujuan organisasi dan harus konsisten dengan pasar, finansial dan tujuan manufaktur. Di dalam mengelola persediaan, ada 3 tujuan utama yang digunakan untuk menilai kinerja dari *inventory control*, antara lain (Arnold, et al., 2008):

- Maksimum *customer service*
- Biaya operasi yang rendah
- Minimum investasi persediaan

Customer service dideskripsikan dengan adanya availabilitas *item* untuk memenuhi permintaan konsumen. *Inventory* digunakan untuk memaksimalkan

customer service dengan mengatasi ketidakpastian pada *demand*. Biasanya, *inventory* yang digunakan untuk mengatasi fluktuasi yang diakibatkan karena *demand* maupun *lead time* disebut dengan *safety stock*.

2.1.1 Biaya Persediaan

Salah satu tujuan perusahaan dalam mengelola *inventory* adalah dengan meminimasi total biaya *inventory*. Kelas biaya dalam *inventory* dibagi menjadi 3, yaitu biaya pengadaan, biaya simpan, dan biaya *stockout* (Ballou, 2004). Berikut merupakan 3 kelas biaya dalam *inventory*:

1. *Procurement Cost*

Biaya pemesanan atau *procurement cost* merupakan biaya yang dikeluarkan sehubungan dengan kegiatan pemesanan barang yang dimulai dari dilakukannya pemesanan hingga tersedianya barang tersebut. Biaya pemesanan ini merupakan *fixed cost* yang tidak tergantung pada jumlah yang dipesan tetapi bergantung pada berapa kali pesanan tersebut dilakukan. Di dalam *procurement cost*, juga terdapat biaya pembelian atau *purchasing cost* yang menyangkut pada biaya yang harus dikeluarkan untuk membeli suatu produk berdasarkan harga dan kuantitas pembelian.

2. *Carrying Cost*

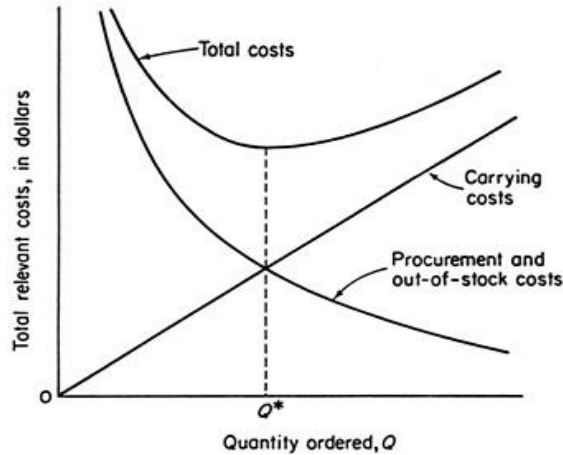
Inventory carrying cost atau biasa disebut *holding cost* merupakan hasil biaya dari penempatan dan penyimpanan dari suatu barang selama periode tertentu di dalam Gudang. Biaya-biaya tersebut meliputi *space cost*, *capital cost*, *inventory service cost*, dan *inventory risk cost*.

3. *Out-of-Stock Cost*

Stockout terjadi ketika jumlah persediaan di gudang tidak dapat memenuhi kebutuhan dari permintaan. Akibat dari adanya *stockout* tersebut, berdampak pada munculnya tambahan biaya seperti *lost sale cost* dan *backorder cost* (Ballou, 2004). *Lost sale cost* merupakan hilangnya *profit* yang seharusnya bisa didapatkan dari penjualan perusahaan karena permintaan dari *customer* tidak bisa dipenuhi. *Lost sale cost* biasanya muncul untuk produk-produk jadi. Sedangkan *backorder cost* muncul ketika *customer* terlambat menerima order, sehingga perlu dilakukan pemesanan kembali dengan pengiriman yang

lebih cepat dari biasanya. *Backorder cost* memiliki biaya yang lebih mahal dari *ordering cost* yang dikenai perusahaan.

Dari ketiga biaya tersebut muncul *trade-off* dalam kaitannya dengan jumlah *item* yang dipesan dalam persediaan. Berikut merupakan grafik yang menunjukkan konflik dari biaya tersebut.



Gambar 2.1 *Trade Off* Komponen Biaya *Inventory*
Sumber: (Ballou, 2004)

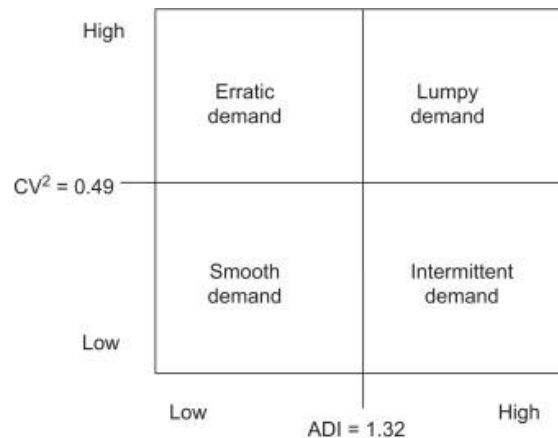
Trade-off dari komponen biaya tersebut menghasilkan total biaya yang minimum. Titik dimana total biaya yang minimum ini akan digunakan sebagai kebijakan *inventory* dalam menentukan kuantitas pesanan yang optimal, atau biasa disebut *Economic Order Quantity* (EOQ).

2.1.2 *Klasifikasi Demand*

Melakukan klasifikasi pada pola permintaan merupakan elemen penting didalam mengelola persediaan. Perbedaan kategori pada pola permintaan tersebut dapat mempengaruhi metode dan kebijakan didalam pengendalian *inventory*. Williams (1984) memberikan gagasan yang disebut *variance partition*, yang membagi variansi permintaan selama *lead time* ke bagian penyusunnya untuk mengidentifikasi metode peramalan dan *inventory control* setiap kategorinya. Kemudian Eaves & Kingsman (2004) mengembangkan ide dari Williams bahwa demand diklasifikasikan berdasarkan 3 elemen, yaitu *variability* dari *transaction rate*, *demand size variability*, dan *lead time variability*.

Kedua gagasan tersebut digunakan dalam memilih metode peramalan yang tepat. Akan tetapi, nilai pemisah setiap kategori *demand* didasarkan pada aspek

kualitatif sehingga menciptakan keraguan pada penerapan skema tersebut di kasus yang berbeda (Heinecke, et al., 2013). Atas dasar alasan tersebut Syntetos et al. (2005) mengusulkan matriks yang ditunjukkan pada Gambar 2.2 dimana nilai *cut-off* secara matematis ditentukan melalui perbandingan *Mean Squared Error*.



Gambar 2.2 Matriks Pembagian 4 Tipe *Demand*
Sumber: (Syntetos AA, 2005)

Matriks pada Gambar 2.2 tersebut dibagi kedalam 4 kuadran yang ditentukan melalui *demand quantity variability*, yang diperoleh dari *squared covariance* (CV^2), dan *average demand interval* (ADI). Nilai *cut-off* yang diperoleh Syntetos et al. (2005) masing-masing adalah 0.49 dan 1.32. Berdasarkan 2 aspek tersebut, kelas *demand* dibagi menjadi *smooth*, *erratic*, *lumpy*, dan *intermittent*.

Misalnya pada jenis permintaan dari *maintenance material* atau *spare part* yang memiliki pola *demand* yang *intermittent* dan *lumpy*. *Spare part* dikatakan *intermittent* atau *sporadic* karena memiliki interval periode setiap permintaan tinggi ($ADI > 1.32$) bergantung pada kebutuhan *maintenance*, dan dikatakan *lumpy* apabila variabilitas antar permintaannya tinggi ($CV^2 > 0.49$).

2.2 Inventory Policy

Terdapat banyak kebijakan didalam mengelola persediaan. Kebijakan ini bergantung dari jenis pola permintaan, kuantitas pesanan, jumlah periode, biaya, dan permasalahan lainnya. Untuk pengelolaan *inventory* secara individu atau *independent item* digunakan pendekatan model EOQ. Model EOQ dapat menentukan berapa jumlah kuantitas pesanan yang optimal untuk meminimasi total biaya *inventory* (Ballou, 2004).

Untuk mengelola *inventory* dengan banyak *item*, digunakan kebijakan *replenishment*. Menurut Ghorbel et al. (2014), terdapat 4 kebijakan persediaan dasar pada *replenishment inventory* yang dibagi ke dalam *continous review system* dan *periodic review system*. *Replenishment policy* yang dimaksud antara lain:

- (s, Q) Policy : Ketika posisi persediaan berada di bawah atau tepat di *reorder point* s , pesanan ditempatkan dengan kuantitas yang tetap Q .
- (s, S) Policy : Ketika posisi persediaan berada di bawah atau tepat di *reorder point* s , order ditempatkan dengan kuantitas tertentu hingga mencapai posisi *order-up-to level* S .
- (T, S) Policy : Posisi persediaan dilakukan *review* pada interval waktu T , dengan setiap *review* menempatkan pesanan dengan kuantitas tertentu hingga mencapai posisi *order-up-to level* S .
- (T, s, S) Policy : Posisi persediaan dilakukan *review* pada interval waktu T , dengan setiap *review* menempatkan pesanan dengan kuantitas tertentu hingga mencapai posisi *order-up-to level* S dan dilihat berdasarkan titik *reorder point* s .

Berikut merupakan tabel pembagian kelas *replenishment policy* berdasarkan kuantitas dan periode yang tetap atau tidak.

Tabel 2.1 *Replenishment Policy* pada *Inventory Control*

Q : Quantity	T : Periodicity	
	Fixed	Variable
Fixed	-	(s,Q)
Variable	(T,S);(T,s,S)	(s,S)

Sumber : (Ghorbel, et al., 2014)

Sipper & Bulfin (1997) juga membagi kebijakan *inventory* berdasarkan adanya permintaan yang *intermittent* atau tidak. Untuk waktu yang kontinu dilakukan kebijakan *continuous review*, yaitu dengan menggunakan model EOQ, (s, Q) policy, *base stock*, atau *two bins inventory*. Apabila waktu antar *demand* tergolong *intermittent* maka dapat dilakukan *periodic review*, yaitu dengan menggunakan EOQ, (s, T) policy, (s, S) policy, atau *optional replenishment / joint replenishment*.

2.2.1 Joint Replenishment Policy

Persediaan tidak hanya dikelola melalui tiap individu *item*. Proses *purchasing* bias dilakukan pada beberapa *item* secara simultan. Hal ini menunjukkan bahwa pada pengelolaan *inventory* bisa dilakukan untuk beberapa grup *item* yang berkorelasi, misalnya *item* yang dibeli pada *supplier* yang sama. Manajemen persediaan dengan mengkoordinasikan jumlah pesanan pada beberapa *item* dalam satu Gudang inilah yang disebut sebagai *joint replenishment policy* (JRP) (Nagasawa & Irohara, 2016). JRP tergolong ke dalam permasalahan *NP-Hard* karena adanya kondisi kombinatorial dalam penyelesaiannya. Porras & Dekker (2008) melakukan pendekatan *single item* untuk persediaan pada 20 produk menggunakan strategi *joint replenishment*. Strategi JRP tersebut dapat menghasilkan penghematan total biaya sebesar 13% jika dibandingkan dengan metode *Economic Order Quantity* (EOQ).

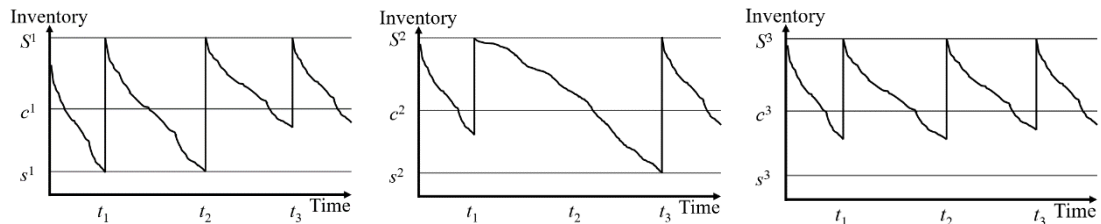
Tujuan dilakukannya pengendalian persediaan dengan JRP untuk *multi-item*, antara lain (Putra & Pujawan, 2011):

1. Menghemat *purchasing cost* per unit. Pembelian dari *item* yang dilakukan dengan kuantitas yang banyak pada kelompok yang sama yaitu dengan *supplier* yang sama dapat memperoleh potongan harga.
2. Menghemat biaya transportasi. *Supplier* yang awalnya mengirimkan *item* secara individu *item* dapat dilakukan dengan mengirim kelompok *item*. Dengan menambah kuantitas dari *item*, biaya tetap yang dikenai *supplier* saat mengangkut barang relative lebih rendah untuk setiap *item*nya.
3. Menghemat biaya pemesanan. Untuk mengurangi biaya pemesanan, *item* dikelompokkan kedalam satu grup untuk dilakukan pemesanan di waktu yang sama (*single order*).
4. Memudahkan penjadwalan. Pengelolaan pesanan yang terkoordinasi akan memberikan kemudahan penjadwalan waktu pemesanan, tugas penerimaan, dan inspeksi.

Berdasarkan *joint order release*, JRP bisa dilakukan dengan beberapa kebijakan *inventory* (Fogarty, et al., 1991), seperti *Order Point System*, (*s, c, S*) *system*, *periodic review system*, dan *Time-Phased Order Point* (TPOP) *System*.

2.3 Can-Order Policy

Can-order policy atau (s, c, S) *policy* adalah salah satu tipe dari *joint replenishment policy*. Balintfy (1964) menunjukkan bahwa *can-order policy* setidaknya lebih baik dari pada *replenishment policy* terkoordinasi lainnya. Menurut Silver (1974), *can order policy* (sistem s, c, S) digunakan untuk mengendalikan *item - item* yang terkoordinasi dengan menggunakan 3 parameter, yaitu *order-up-to level* (S), *can-order level* (c), dan *must-order level* atau *re-order level* (s). Ketika level persediaan dari suatu *item* berada dibawah s , pesanan ditempatkan pada level persediaan *item* tersebut mencapai S . Untuk *item* lain yang level persediannya dibawah c , maka pesanan akan ditempatkan pada level persediaan mencapai S . Berikut merupakan gambaran dari persediaan menggunakan *can order policy*.



Gambar 2.3 Konvensional *Can-Order Policy* pada 3 *item*
Sumber : (Nagasawa & Irohara, 2016)

Sistem (s, c, S) dapat menurunkan biaya pesanan dengan mengeleminasi biaya pesanan minor yang dihubungkan dengan *item* yang tidak dilakukan pesanan, dan dapat mereduksi *inventory carrying cost* dengan mereduksi kebutuhan stok di gudang (Fogarty, et al., 1991). Tetapi, biaya *inventory* bisa saja naik jika interval antara *joint order* menurun karena adanya *item* yang terpicu untuk dipesan pada saat waktunya belum dipesan.

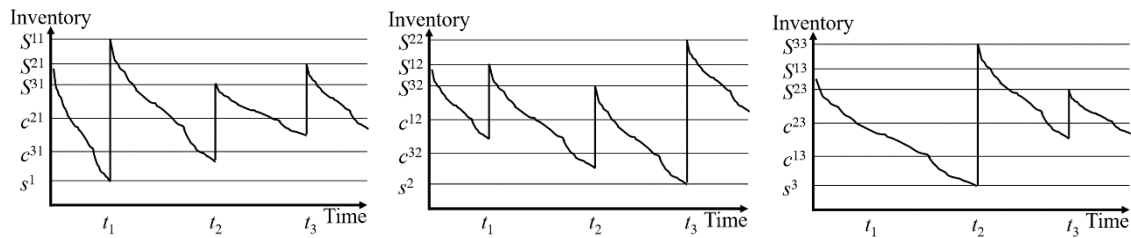
Federgruen, et al. (1984) memodelkan *can-order policy* dengan *semi-Markov* yang berisi *poisson demand* dan positif *lead time*. Mereka menunjukkan bahwa *can-order policy* memiliki hasil yang lebih baik dari sisi biaya sebesar 20% dibandingkan *uncoordinated policy*.

2.3.1 Multiple Can-order Level pada Can-order Policy

Multiple can-order level pada kebijakan *can-order* diusulkan pertama kali oleh Nagasawa dan Irohara pada tahun 2016. Nagasawa (2016) mengadopsi model

pada *ordering policy* yang mempertimbangkan adanya korelasi *item*. *Correlated demand* merupakan adanya pengaruh dari permintaan dari 2 atau lebih *item* pada sekali pembelian dengan probabilitas tinggi. Dari model *can-order policy*, perubahan dilakukan pada *can-order level* dan *order-up-to level* bergantung pada *ordering trigger item*.

Pada model Nagasawa (2016), untuk setiap *item i*, *ordering policy* yang dilakukan berupa (s_i, c_{ij}, S_{ij}) . Ketika *inventory level* dari *item i* berada pada *re-order level* s_i , hal tersebut akan memicu *replenishment order* yang menaikkan level persediaan *item i* mencapai *order-up-to level* dari *item* tersebut S_{ii} . Pada waktu yang sama, jika level persediaan dari *item k* lainnya berada dibawah atau tepat pada *can-order level* ketika *item i* berada pada *re-order level*, lalu *can-order level*-nya disebut sebagai c_{ik} . *Item k* dimasukkan kedalam tambahan pesanan mencapai *order-up-to level* untuk *item i*, S_{ik} . Berikut merupakan gambaran *can-order policy* dengan *multiple can-order level*.



Gambar 2.4 *Multiple Can-order Level* pada Sistem (s, c, S) untuk 3 *item* yang berkorelasi

Sumber : (Nagasawa & Irohara, 2016)

Model yang dibangun dilakukan pada permintaan deterministik yang telah diketahui selama *planning horizon*. Tujuan dari permasalahan ini adalah untuk mencari waktu dan kuantitas *replenishment* yang optimal. Model yang dikembangkan juga didasarkan pada asumsi-asumsi, seperti:

- *Supplier* menjual produknya tanpa dikenakan diskon
- *Replenishment item* diantarkan dengan kapasitas angkut / *carrier* yang tak terbatas
- *Item* yang disimpan merupakan *item non-perishable* yang tidak deteriorasi.
- Ketika permintaan tidak terpenuhi (*stockout*), maka akan dikenakan *lost-sale penalty*.

Model matematis *integer non linear programming* (INLP) yang dikembangkan oleh Nagasawa (2016), adalah sebagai berikut :

- Parameter :

i = jenis *item* ($i \in I$)

j = jenis *correlated item* ($j \in I$)

t = waktu perencanaan ($t \in T$)

M = big-M

d_{it} = *demand item* i pada periode ke t

v_i = *volume item* i

b = kapasitas angkut

- Variabel Keputusan :

y_t = jumlah angkutan pada periode ke t

s_i = *re-order level item* i

c_{ij} = *can-order level item* j ketika *item* i dipesan

S_{ij} = *order-up-to level item* j ketika *item* i dipesan

o_{it} = jumlah *item* i yang *stockout* pada periode t

I_{it} = *inventory level item* i pada periode t

x_{it} = jumlah *order quantity item* i pada periode t

$r_{it} = \begin{cases} 1, & \text{jika lit berada di bawah } s_i \text{ pada periode } t \\ 0, & \text{otherwise} \end{cases}$

$k_{ijt} = \begin{cases} 1, & \text{jika lit berada di bawah } c_{ij} \text{ pada periode } t \\ 0, & \text{otherwise} \end{cases}$

$q_{ijt} = \begin{cases} 1, & \text{jika lit berada di bawah } c_{ij}, \\ & \text{setidaknya 1 item dipesan pada} \\ & \text{periode } t \text{ dan tambahan pesanan} \\ & \text{ditempatkan di } S_{ij} \\ 0, & \text{otherwise} \end{cases}$

- Fungsi Tujuan

$$\text{Min} \quad Z_1 = \sum_{t=1}^T y_t \quad (2.1)$$

$$\text{Min} \quad Z_2 = \sum_{t=1}^T \sum_{i=1}^I o_{it} \quad (2.2)$$

$$\text{Min} \quad Z_3 = \sum_{t=1}^T \sum_{i=1}^I I_{it} \quad (2.3)$$

- Konstrain

$$I_{i,t-1} + x_{it} - I_{it} + o_{it} = d_{it}; \forall i, \forall t \quad (2.4)$$

$$\sum_{i=1}^I v_i x_{it} \leq by_t; \forall t \quad (2.5)$$

$$I_{i,t-1} - d_{it} + o_{it} + Mr_{it} \geq s_i; \forall i, \forall t \quad (2.6)$$

$$I_{i,t-1} - d_{it} + o_{it} + M(1 - r_{it}) \leq s_i; \forall i, \forall t \quad (2.7)$$

$$S_{ii} \leq I_{i,t-1} - d_{it} + o_{it} + x_{it} + M(1 - r_{it}); \forall i, \forall t \quad (2.8)$$

$$S_{ii} \geq I_{i,t-1} - d_{it} + o_{it} + x_{it} + M(1 - r_{it}); \forall i, \forall t \quad (2.9)$$

$$S_{ij} \geq c_{ij}; \forall i, \forall j \quad (2.10)$$

$$c_{ij} \geq s_j; \forall i, \forall j \quad (2.11)$$

$$I_{j,t-1} - d_{jt} + o_{jt} + Mk_{ijt} \geq c_{ij}; \forall i, \forall j, \forall t \quad (2.12)$$

$$I_{j,t-1} - d_{jt} + o_{jt} - M(1 - k_{ijt}) \leq c_{ij}; \forall i, \forall j, \forall t \quad (2.13)$$

$$q_{ijt} \leq k_{ijt}; \forall i, \forall j, \forall t \quad (2.14)$$

$$q_{ijt} \leq r_{it}; \forall i, \forall j, \forall t \quad (2.15)$$

$$\sum_{i=1}^I q_{ijt} \leq 1; \forall i, \forall t \quad (2.16)$$

$$c_{ij} k_{ijt} \leq \sum_{i=1}^I c_{ij} q_{ijt}; \forall i, \forall j, \forall t \quad (2.17)$$

$$S_{ij} \leq I_{j,t-1} - d_{jt} + o_{jt} + x_{jt} + M(1 - q_{ijt}); \forall i, \forall j, \forall t \quad (2.18)$$

$$S_{ij} \geq I_{j,t-1} - d_{jt} + o_{jt} + x_{jt} - M(1 - q_{ijt}); \forall i, \forall j, \forall t \quad (2.19)$$

$$x_{jt} \leq M(r_{jt} + \sum_{i=1}^I q_{ijt}; \forall j, \forall t) \quad (2.20)$$

$$o_{it} \leq d_{it}; \forall i, \forall t \quad (2.21)$$

$$r_{it}, k_{ijt}, q_{ijt} = \{0,1\}; \forall i, \forall j, \forall t \quad (2.22)$$

$$y_t, I_{it}, x_{it}, o_{it}, s_i, c_{ij}, S_{ij} \geq 0; \forall i, \forall t \quad (2.23)$$

Fungsi tujuan yang pertama pada persamaan 2.1 adalah jumlah dari angkutan yang dipakai selama waktu perencanaan. Fungsi tujuan kedua (2.2) adalah jumlah dari total *item shortage* selama waktu perencanaan. Fungsi tujuan ketiga (2.3) adalah total dari persediaan di gudang selama waktu perencanaan.

Sementara itu, terdapat kendala-kendala yang digunakan dalam model. Konstrain 2.4 merupakan persamaan keseimbangan persediaan untuk semua *item*. Konstrain 2.5 adalah konstrain pada pesanan *item* selama periode waktu. Pada konstrain 2.6 dan 2.7, menunjukkan jika level persediaan jatuh pada atau di bawah *re-order level*, maka pesanan akan ditempatkan. Konstrain 2.8 dan 2.9 adalah ketika level persediaan persediaan jatuh pada atau di bawah *re-order level*, kuantitas pesanan ditentukan dari selisih *order-up-to level* dan *inventory level*. Konstrain 2.10 menunjukkan bahwa setiap *can-order level* tidak lebih dari nilai *order-up-to level*. Sama halnya pada konstrain 2.11 yang menunjukkan bahwa *re-order level* tidak lebih dari *can-order level*. Konstrain 2.12 dan 2.13 mengidentifikasi apakah level persediaan dari suatu *item* berada pada atau di bawah *can-order level* dari *item* tersebut. Pada konstrain 2.14 sampai 2.17, menjelaskan ketika level persediaan dari setidaknya satu *item* berada pada atau di bawah *re-order level*, pesanan ditempatkan termasuk *item* yang level persediaannya berada dibawah atau pada *can-order level*. *Can-order level* ditetapkan menjadi maksimum *can-order level* dari *item* yang terpicu. Pada konstrain 2.18 dan 2.19, *item* yang level persediaannya berada dibawah atau pada *can-order level*, yang berhubungan dengan *item* pemicu lainnya, dipesan dalam kuantitas sebesar selisih dari level persediaan dan *order-up-to level* yang memiliki hubungan dengan *item* lain terkorelasi. Konstrain 2.20 menunjukkan bahwa *item* dengan level persediaan berada pada atau di bawah *can-order* dan *re-order level* bisa dilakukan pemesanan. Konstrain 2.21 menunjukkan bahwa jumlah

item yang *shortage* pada setiap *item* tidak lebih dari permintaan pada periode tersebut. Konstrain 2.22 merupakan konstrain untuk variabel biner dan konstrain 2.23 merupakan konstrain *non-negativity* pada variabel.

2.4 Algoritma *Simulated Annealing* untuk Optimasi

Metaheuristik merupakan metoda untuk mencari solusi yang memadukan interaksi antara prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari lokal optimal (Santosa & Willy, 2011).

Salah satu metode pada metaheuristik yang meniru peroses fisik adalah *Simulated Annealing* (SA). Algoritma SA pertama kali ditemukan oleh Metropolis, Rosenbluth dan Teller pada tahun 1953 untuk menyimulasikan proses pendinginan material dalam *heat bath*. *Annealing* adalah satu teknik yang dikenal dalam bidang metalurgi, digunakan dalam mempelajari proses pembentukan kristal dalam suatu materi (Santosa & Willy, 2011). Agar dapat terbentuk susunan kristal yang sempurna, diperlukan pemanasan sampai suatu tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Pemanasan materi di awal proses *annealing*, memberikan kesempatan pada atom-atom dalam materi itu untuk bergerak secara bebas, mengingat tingkat energi dalam kondisi panas ini cukup tinggi. Proses pendinginan yang perlahan-lahan memungkinkan atom-atom yang tadinya bergerak bebas itu, pada akhirnya menemukan tempat yang optimum, di mana energi internal yang dibutuhkan atom itu untuk mempertahankan posisinya adalah minimum.

Pada awal proses SA, dipilih suatu solusi awal, yang merepresentasikan kondisi materi sebelum proses dimulai. Gerakan bebas dari atom-atom pada materi, direpresentasikan dalam bentuk modifikasi terhadap solusi awal/solusi sementara. Pada awal proses SA, saat parameter suhu (T) diatur tinggi, solusi sementara yang sudah ada diperbolehkan untuk mengalami modifikasi secara bebas.

Kirkpatrick, Gellat, dan Vecchi mengembangkan sebuah algoritma untuk mengaplikasikan algoritma ini pada permasalahan optimisasi pada tahun 1983. Dalam SA terdapat konsep probabilitas yang disebut dengan distribusi probabilitas Boltzman, dengan rumus sebagai berikut :

$$P(E) = e^{-E/kT} \quad (2.24)$$

dimana,

$P(E)$ = peluang untuk mencapai tingkat energi E

T = temperatur

k = konstanta Boltzman

Dalam penerapannya pada konteks minimasi fungsi, solusi adalah x dan nilai fungsinya adalah $f(x)$, dimana $f(x)$ ini mirip dengan status energi pada sistem termodinamika, dengan rumus sebagai berikut :

$$E_i = f_i = f(x_i) \quad (2.25)$$

Menurut kriteria metropolis, probabilitas penerimaan titik solusi selanjutnya yakni x_{i+1} bergantung pada perbedaan status energi atau fungsi tujuan yang didapatkan sebagai berikut :

$$P[E_{i+1}] = \min\{1, e^{-E/kT}\} \quad (2.26)$$

dimana,

$$\Delta E = E_{i+1} - E_i = \Delta f = f_{i+1} - f_i = f(x_{i+1}) - f(x_i) \quad (2.27)$$

Distribusi probabilitas Boltzmann ini digunakan sebagai acuan untuk menerima terlebih dahulu solusi yang lebih buruk atau tidak sehingga dapat menghindari jebakan *local optima*. Nilai k merupakan faktor Boltzmann yang biasanya bernilai 1. Dalam kasus minimasi jika $\Delta E \leq 0$, maka $P[E_{i+1}] = 1$ maka titik x_{i+1} akan diterima. Tetapi, jika $\Delta E > 0$, maka x_{i+1} belum tentu ditolak karena masih dimungkinkan untuk diterima dengan probabilitas Boltzmann seperti formulasi di bawah ini:

$$P(E_{i+1}) = e^{-E/kT} \quad (2.28)$$

Pada persamaan di atas dapat diinterpretasikan bahwa ketika temperatur T masih tinggi maka peluang menerima x_{i+1} dengan ΔE yang besar akan lebih besar, dan dengan semakin menurunnya nilai T maka peluang menerima x_{i+1} yang lebih buruk akan semakin kecil, sehingga pencarian solusi dapat mencapai titik konvergensinya. Nilai T akan semakin menurun dikarenakan terdapat faktor pereduksi temperatur, c , di tiap siklus iterasinya. Jika hasil dari solusi vektor baru lebih baik dibandingkan solusi awal maka terima solusi baru, namun jika tidak

gunakan kriteria metropolis untuk menentukan apakah solusi baru diterima atau tidak.

2.5 Optimasi *Multi-Objective*

Optimasi multitujuan merupakan persoalan optimasi yang memiliki lebih dari satu fungsi tujuan, yang memungkinkan terjadi konflik antar fungsi tujuan tersebut (Santosa & Willy, 2011). Cohon & Marks (1975) menjelaskan klasifikasi teknik yang populer di *Operations Research* yang fokus untuk mengatasi beberapa permasalahan fungsi tujuan sebagai berikut :

- *A Priori Preference*: pengambilan keputusan dilakukan sebelum pencarian fungsi tujuan. Teknik ini melakukan pendekatan dengan beberapa kemungkinan pencapaian tujuan yang dilakukan *decision maker*. Metode yang digunakan antara lain, *goal programming* dan *global criterion*.
- *A Posteriori Preference*: pencarian tujuan dilakukan sebelum mengambil keputusan. Teknik ini dilakukan apabila pengambil keputusan tidak memiliki preferensi terhadap permasalahan yang dihadapi.
- *Progressive Preference*: integrasi antara pencarian dan pengambilan keputusan. Teknik ini dilakukan dengan 3 tahapan yaitu, mencari *non-dominated solution*, memodifikasi preferensi berdasarkan solusi non-dominasi, dan memodifikasi preferensi kembali hingga tidak ada perubahan keputusan.

2.5.1 Metode *Global Criterion*

Global kriteria merupakan salah satu dari pendekatan *a priori* yang menjelaskan teknik untuk permasalahan fungsi tujuan dengan menjadikannya sebagai fungsi tunggal. Tujuannya adalah untuk mengukur seberapa dekat pengambil keputusan dapat menentukan solusi optimal. Fungsi *multi objective* yang digunakan pada global kriteria (Tabucanon, 1988), adalah sebagai berikut.

$$\text{Min} \quad F = \sum_{l=1}^K \left[\frac{f_l(x^*) - f_l(x)}{f_l(x^*)} \right]^p \quad (2.29)$$

dimana,

$f_l(x)$ = fungsi tujuan l

$f_l(x^*)$ = Nilai optimal pada fungsi objektif l (solusi fungsi ideal)

p = Bobot terhadap nilai penyimpangan

Misalnya, jika $p = 1$, maka besar kecilnya nilai penyimpangan diberi nilai yang sama menjadi fungsi *linear*. Jika $p > 1$, maka fungsi akan mendahulukan penyimpangan yang lebih besar sebagai *entering variable* menjadi fungsi *quadratic*. Besarnya nilai p bergantung pada pengambil keputusan karena hasil solusi bisa saja menjadi tidak diterima. Umumnya, nilai p yang digunakan sebesar 1 dan 2. Nilai $p = 2$, digunakan sebagai dasar penggunaan perhitungan jarak *Euclidean*.

Keuntungan dari metode ini adalah simpel untuk digunakan dan efektif karena tidak perlu dilakukan *pe-ranking-an* secara Pareto oleh pengambil keputusan. Sedangkan kekurangannya adalah definisi dari fungsi tujuan perlu dilakukan komputasi yang cukup rumit dan pengambil keputusan tidak bisa memberikan subjektivitas pada model.

2.6 Posisi Penelitian

Penelitian tentang penyelesaian permasalahan *inventory* menggunakan *can-order policy* terjadi banyak perkembangan. Kebanyakan penelitian tersebut focus pada bagaimana mendapatkan nilai (s, c, S) yang optimal untuk setiap *item*, seperti yang dikembangkan Atkins & Iyogun (1988) yang menggunakan algoritma heuristik untuk meminimasi total biaya persediaan. Begitu pula Johansen & Melchior (2003), yang menggunakan *Markov Decision Theory* dengan mempertimbangkan permintaan yang stokastik. Kemudian untuk mempertimbangkan adanya korelasi permintaan, Tsai et al. (2009) mengembangkan *clustering algorithm* untuk mengevaluasi adanya korelasi *demand*. Mereka menggunakan *support concept* untuk mengukur *similarity* antar *item*.

Kebijakan *can-order* juga dapat dilakukan simulasi dengan mempertimbangkan ketidakpastian pada *lead time* dan *demand*. Hal tersebut yang dilakukan Fransiska (2012) untuk mengendalikan *inventory* komponen *circuit breaker* dan Putra & Pujawan (2011) untuk mengendalikan persediaan *spare part* pada pembangkit. Masing-masing menggunakan simulasi diskrit dan simulasi Monte Carlo.

Fungsi tujuan yang digunakan dalam kebanyakan penelitian tersebut adalah untuk meminimasi total biaya persediaan. Dari semua penelitian tersebut, masih belum ada yang mempertimbangkan kapasitas angkut pada *supplier*. Untuk mengembangkan model yang juga memperhatikan adanya korelasi pada *demand*, Nagasawa (2016) mengembangkan model *multiple can-order level*, yang memperhatikan adanya kapasitas angkut. Selain itu, model yang dikembangkan berupa fungsi *multi objective*, untuk meminimasi *carrier*, level persediaan, dan jumlah *item stockout* dengan penyelesaian menggunakan *Genetic Algorithm*.

Sehingga, pada penelitian ini, penulis mengembangkan model metaheuristik lain dengan *neighborhood based* untuk memudahkan proses komputasi dengan *Simulated Annealing* dan metode global kriteria pada penyelesaian fungsi multi tujuan.

Tabel 2.2 Posisi Penelitian

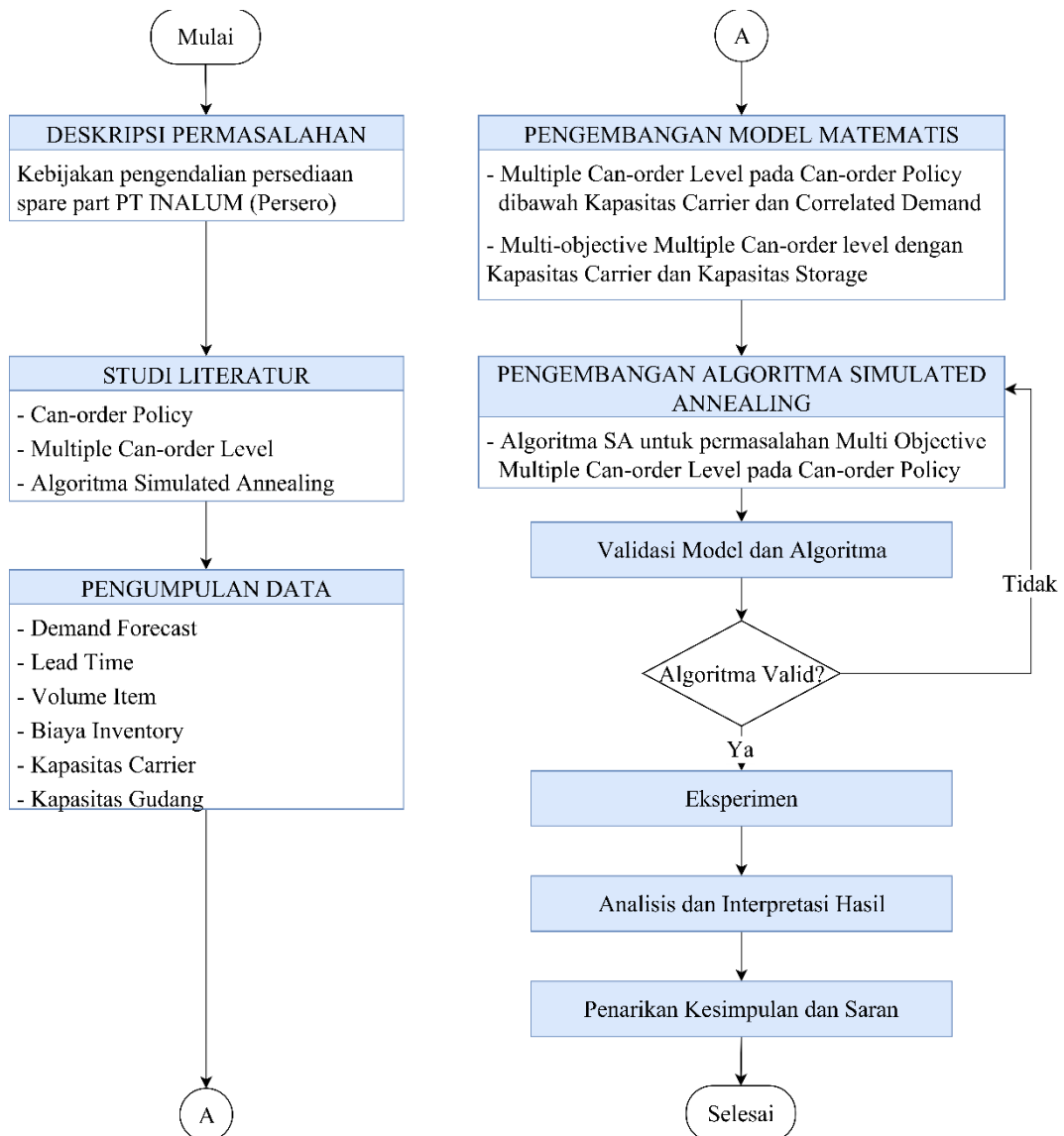
Penelitian	Multiple COL	Lead Time	Demand	Carrier Capacity	Multi Tujuan	Fungsi Tujuan	Metode Penyelesaian
Atkins & Iyogun (1988)		Fixed	Deterministik			Minimasi total biaya	<i>Heuristic</i>
Johansen & Melchiors (2003)		Fixed	Stokastik			Minimasi total biaya	<i>Markov Decision Theory</i>
Tsai et al. (2009)		Fixed	Deterministik			Minimasi total biaya	<i>Association Clustering Algorithm</i>
Fransiska (2011)		Variable	Stokastik			Minimasi total biaya	Simulasi Diskrit
Putra & Pujawan (2011)		Variable	Stokastik			Minimasi total biaya	Simulasi Monte Carlo
Nagasawa (2016)	√	Fixed	Deterministik	√	√	Minimasi <i>carrier</i> , Minimasi level persediaan, Minimasi jumlah <i>stockout</i>	<i>Genetic Algorithm</i>
Penelitian ini	√	Fixed	Deterministik	√	√	Minimasi total biaya & Minimasi jumlah <i>carrier</i>	<i>Simulated Annealing</i>

(Halaman Ini Sengaja Dikosongkan)

BAB 3

METODOLOGI PENELITIAN

Pada bagian ini akan dijelaskan mengenai tahapan-tahapan dalam melakukan penelitian mengenai pengembangan algoritma *simulated annealing* pada permasalahan *multiple can-order level* dengan tujuan lebih dari satu. Langkah-langkah metodologi yang dilakukan dapat dilihat pada *flowchart* Gambar 3.1 berikut.

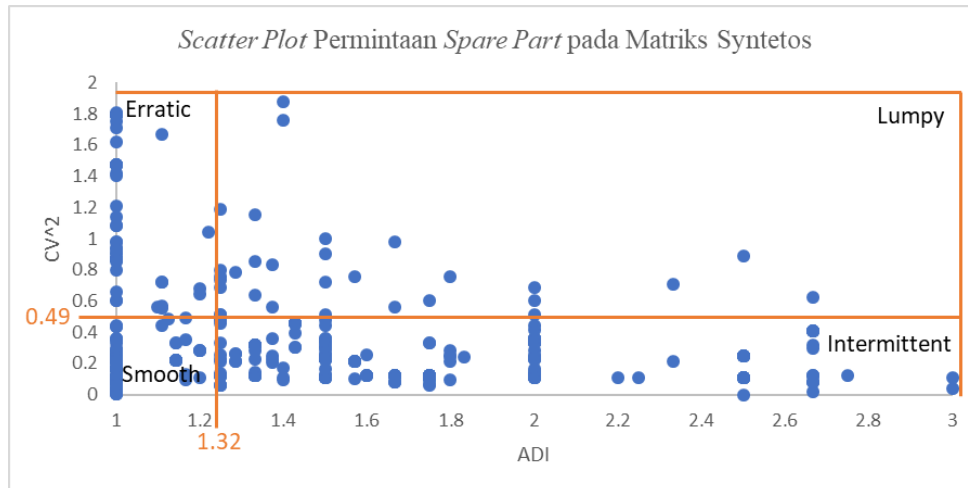


Gambar 3.1 *Flowchart* Metodologi Penelitian

3.1 Deskripsi Permasalahan

Deskripsi permasalahan dilakukan untuk mengetahui kondisi sebenarnya tentang penerapan kebijakan persediaan di perusahaan. Pendalaman permasalahan dilakukan di *Spare Part Warehouse* PT Indonesia Asahan Aluminium (Persero) atau PT. INALUM (Persero)

PT INALUM (Persero) merupakan salah satu pabrik peleburan aluminium terbesar di Asia Tenggara. Untuk melakukan proses produksi dari bahan baku sampai produk jadi, perusahaan membutuhkan kinerja dari mesin-mesin produksi. Sehingga, keandalan dari mesin maupun *supply tools* dalam proses produksi merupakan salah satu faktor penting agar *demand* dari produk jadi tetap bisa dipenuhi seutuhnya. Untuk menjaga keandalan dari peralatan produksi tersebut, perusahaan melakukan strategi *maintenance*, seperti *preventive maintenance* maupun *corrective maintenance*. Untuk melakukan *preventive maintenance*, dibutuhkan adanya *spare part*, yang mana *spare part* tersebut merupakan komponen-komponen pendukung dari mesin-mesin produksi. Saat *spare part* tersebut tidak tersedia, maka proses *maintenance* yang seharusnya dilakukan menjadi tidak dapat dilakukan sampai *spare part* dari mesin tersebut tersedia. Hal ini dapat mengakibatkan berhentinya proses produksi, yang mengakibatkan *demand* dari *customer* tidak bisa terpenuhi. Hal tersebut menunjukkan bahwa pentingnya pengendalian *inventory* pada *spare part*. PT INALUM (Persero) mengendalikan *spare part inventory* sebanyak kurang lebih 14.000 jenis *item* yang terdiri dari *mechanical part*, *electrical part*, fabrikasi lokal, material mekanik, *general goods*, *tools* dan pelumas. Dari *item* tersebut diklasifikasikan juga berdasarkan tingkat *criticality*. *Scatter plot* yang menunjukkan bahwa mayoritas permintaan *spare part* di PT INALUM adalah *intermittent* dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Scatter Plot Permintaan Spare Part PT INALUM*
(Sumber: Data *Forecast Kebutuhan Spare Part Inventory Tahun 2018*)

Banyaknya proporsi permintaan yang tergolong *intermittent*, *lumpy*, maupun *erratic* ini menyebabkan kebijakan persediaan tidak bisa dilakukan dengan EOQ, maupun perhitungan *safety stock* pada umumnya karena tidak berdistribusi normal. Jika dilakukan kebijakan tersebut, biaya persediaannya termasuk biaya *safety stock* akan naik, karena adanya variabilitas yang sangat tinggi diakibatkan karena adanya 0 permintaan pada beberapa periode.

Pemesanan *spare part* yang dilakukan secara independen pada PT INALUM (Persero) menyebabkan tingginya *ordering cost*. Padahal, proses tender pada *spare part* tidak terlalu signifikan berpengaruh, karena harus melakukan pembelian *item* yang banyak jenisnya dan penawaran dilihat berdasarkan kesesuaian spesifikasi *part*. Jadi, saat ada *item* dengan *supplier* yang sama, sementara kebutuhannya pada periode yang berbeda, *item – item* tersebut akan dipesan secara independen pada periode dimana *item* tersebut dibutuhkan. Misalnya saja, pada Vendor 1 yang dipilih untuk memenuhi *demand* pada 136 *item* dan Vendor 2 yang memenuhi 103 *item* pada *spare part*. Berikut merupakan total *item* yang dikirimkan oleh Vendor 1 dan 2 pada setiap periode di tahun 2018.

Tabel 3.1 Jumlah *Item* yang Dipenuhi Vendor 1 dan 2 pada Tahun 2018

Periode	1	2	3	4	5	6	7	8	9	10	11	12
Vendor 1	74	41	75	134	416	31	83	12	380	35	6	0
Vendor 2	142	66	93	137	111	65	106	15	6	19	12	0

(Sumber: Data *Forecast Kebutuhan Spare Part Inventory Tahun 2018*)

Jika dilihat pada tabel tersebut, pemesanan kepada vendor selalu dilakukan di setiap periode sesuai dengan kebutuhan *item* di periode tersebut. Vendor 1 paling banyak bisa mengantarkan *item* pada periode 5 sebanyak 416 *item*. Sementara itu, pada periode 6, 7, dan 8 dilakukan pesanan secara independen. Untuk meminimasi biaya pesanan, seharusnya pemesanan bisa dilakukan pada periode ke-6 untuk memenuhi kebutuhan periode 6, 7, dan 8. Sama halnya pada Vendor 2, yang dapat menyatukan pesannya di periode ke-8 untuk memenuhi permintaan di periode 8, 9, 10, dan 11. Hal ini bisa dilakukan dengan meminimasi jumlah angkutan yang dibutuhkan oleh *supplier*.

Selain itu, adanya biaya *inventory* lainnya seperti *holding cost* harus dipertimbangkan. Karena jika dilakukan terlalu banyak pesanan di waktu yang sama, *holding cost* yang dikenai menjadi tinggi. Adanya kapasitas *receiving area* juga menjadi salah satu kendala yang menentukan berapa maksimum *item* yang harus dikirimkan pada satu periode.

3.2 Studi Literatur

Tahapan studi literatur bertujuan untuk mencari referensi terkait dengan kebijakan pengendalian *inventory* pada *spare part* termasuk *can-order policy* dan metode penyelesaiannya yang menggunakan *simulated annealing* dan *global criterion*. Pencarian referensi juga dilakukan untuk mendapatkan model matematis berupa *multiple can-order level* pada *can-order policy*.

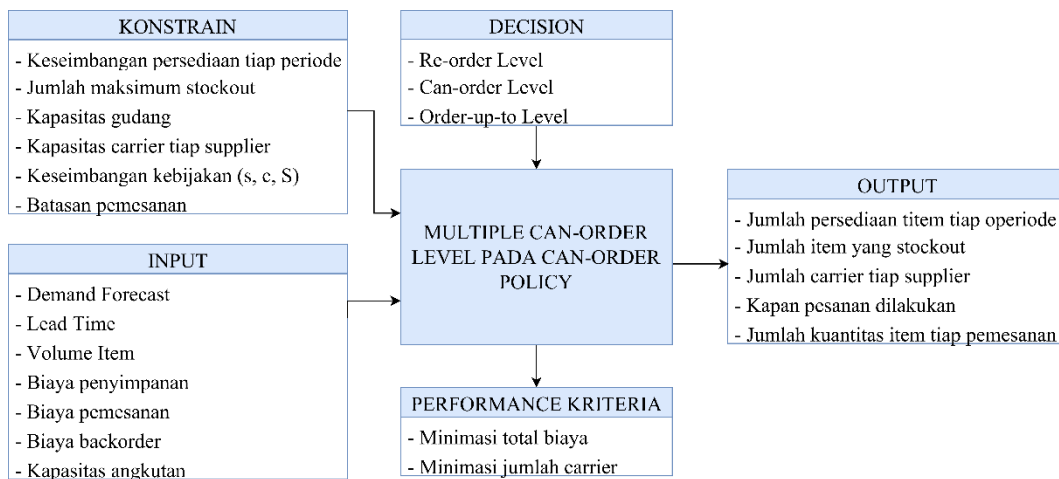
3.3 Pengumpulan Data

Setelah dilakukan analisis permasalahan pada pengendalian persediaan di PT INALUM (Persero), penelitian dilanjutkan dengan pengumpulan data yang dibutuhkan untuk menyelesaikan permasalahan *joint replenishment* dengan *can-order policy*. Data yang dibutuhkan antara lain, jumlah permintaan tiap *item*, *lead time* tiap *supplier*, volume setiap *item*, harga *item*, kapasitas angkut *supplier*, kapasitas gudang, dan biaya-biaya *inventory* yang dikenakan setiap *item*.

3.4 Tahap Pengembangan Model Matematis

Tahap pengembangan model matematis ini dilakukan berdasarkan model yang telah dikembangkan oleh Nagasawa (2016). Model yang dikembangkan merupakan model *multiple can-order level* pada *can-order policy* dengan fungsi tujuan untuk meminimasi penalty.

Dari model tersebut, akan dilakukan modifikasi terhadap penyesuaiannya dengan kondisi riil di perusahaan dengan fungsi tujuan meminimasi total biaya persediaan dan jumlah angkutan *supplier*, dan tambahan kendala kapasitas gudang serta jumlah maksimum *stockout* tiap *item*. Berikut merupakan model sistem pada permasalahan ini.



Gambar 3.3 Model Sistem *Multiple Can-Order Level* pada *Can-Order Policy*

Pada model tersebut dapat diketahui apa saja *input* di dalam model dan juga kendala-kendala yang digunakan. Untuk mencapai kedua fungsi tujuan tersebut, akan ditentukan berapakah nilai *re-order level*, *can-order level*, dan *order-up-to level* yang optimal. Dari solusi yang dibangun, akan muncul *output* seperti level persediaan, kapan dilakukan pesanan, berapa jumlah pesanan yang dilakukan, dan lain-lain. Berikut merupakan model matematis yang dikembangkan pada permasalahan ini :

- Parameter :

i = jenis *item* ($i \in I$)

j = jenis *correlated item* ($j \in I$)

t = waktu perencanaan ($t \in T$)

M = big-M

d_{it} = demand item i pada periode ke t
 v_i = volume item i
 p_i = harga item i
 Ch_i = holding cost item i
 Co_i = orderring cost item i
 Cb_i = back-orderring cost item i
 b = kapasitas angkut
 G = storage capacity

- Variabel Keputusan :

y_t = jumlah angkutan pada periode ke t
 s_i = re-order level item i
 c_{ij} = can-order level item j ketika item i dipesan
 S_{ij} = order-up-to level item j ketika item i dipesan
 o_{it} = jumlah item i yang stockout pada periode t
 I_{it} = inventory level item i pada periode t
 x_{it} = jumlah order quantity item i pada periode t
 $r_{it} = \begin{cases} 1, & \text{jika lit berada di bawah } s_i \text{ pada periode } t \\ 0, & \text{otherwise} \end{cases}$
 $k_{ijt} = \begin{cases} 1, & \text{jika lit berada di bawah } c_{ij} \text{ pada periode } t \\ 0, & \text{otherwise} \end{cases}$
 $q_{ijt} = \begin{cases} 1, & \text{jika lit berada di bawah } c_{ij}, \\ & \text{setidaknya 1 item dipesan pada} \\ & \text{periode } t \text{ dan tambahan pesanan} \\ & \text{ditempatkan di } S_{ij} \\ 0, & \text{otherwise} \end{cases}$

- Fungsi Tujuan :

Minimasi Total *Carrier*:

$$\text{Min} \quad Z_1 = \sum_{t=1}^T y_t \quad (3.1)$$

Minimasi Total Biaya :

Total Biaya = Total *Purchasing Cost* + Total *Holding Cost*
+ Total *Ordering Cost* + Total *Backorderring Cost*

$$\text{Min } Z_2 = \sum_{t=1}^T \sum_{i=1}^I x_{it} p_i + \sum_{t=1}^T \sum_{i=1}^I I_{it} Ch_i + \sum_{t=1}^T \sum_{i=1}^I r_{it} Co_i + \sum_{t=1}^T \sum_{i=1}^I o_{it} Cb_i \quad (3.2)$$

- Konstrain :

Keseimbangan persediaan untuk semua *item* :

$$I_{i,t-1} + x_{it} - I_{it} + o_{it} = d_{it}; \forall i, \forall t \quad (3.3)$$

Jumlah pesanan *item* selama periode waktu dibawah kapasitas angkut :

$$\sum_{i=1}^I v_i x_{it} \leq by_t; \forall t \quad (3.4)$$

Jika level persediaan jatuh pada atau di bawah *re-order level*, maka pesanan akan ditempatkan :

$$I_{i,t-1} - d_{it} + o_{it} + Mr_{it} \geq s_i; \forall i, \forall t \quad (3.5)$$

$$I_{i,t-1} - d_{it} + o_{it} + M(1 - r_{it}) \leq s_i; \forall i, \forall t \quad (3.6)$$

Kuantitas pesanan ditentukan dari selisih *order-up-to level* dan *inventory level* :

$$S_{ii} \leq I_{i,t-1} - d_{it} + o_{it} + x_{it} + M(1 - r_{it}); \forall i, \forall t \quad (3.7)$$

$$S_{ii} \geq I_{i,t-1} - d_{it} + o_{it} + x_{it} - M(1 - r_{it}); \forall i, \forall t \quad (3.8)$$

Can-order level tidak lebih dari nilai *order-up-to level* :

$$S_{ij} \geq c_{ij}; \forall i, \forall j \quad (3.9)$$

Re-order level tidak lebih dari *can-order level* :

$$c_{ij} \geq s_j; \forall i, \forall j \quad (3.10)$$

Level persediaan dari suatu *item* apakah berada pada atau di bawah *can-order level* dari *item* :

$$I_{j,t-1} - d_{jt} + o_{jt} + Mk_{ijt} \geq c_{ij}; \forall i, \forall j, \forall t \quad (3.11)$$

$$I_{j,t-1} - d_{jt} + o_{jt} - M(1 - k_{ijt}) \leq c_{ij}; \forall i, \forall j, \forall t \quad (3.12)$$

Ketika level persediaan dari setidaknya satu *item* berada pada atau di bawah *re-order level*, pesanan ditempatkan termasuk *item* yang level persediaannya berada dibawah atau pada *can-order level* :

$$q_{ijt} \leq k_{ijt}; \forall i, \forall j, \forall t \quad (3.13)$$

$$q_{ijt} \leq r_{it}; \forall i, \forall j, \forall t \quad (3.14)$$

$$\sum_{i=1}^I q_{ijt} \leq 1; \forall i, \forall t \quad (3.15)$$

$$c_{ij}k_{ijt} \leq \sum_{i=1}^I c_{ij}q_{ijt}; \forall i, \forall j, \forall t \quad (3.16)$$

Kuantitas *item* yang dipesan sebesar selisih dari level persediaan dan *order-up-to level* yang memiliki hubungan dengan *item* lain terkorelasi :

$$S_{ij} \leq I_{j,t-1} - d_{jt} + o_{jt} + x_{jt} + M(1 - q_{ijt}); \forall i, \forall j, \forall t \quad (3.17)$$

$$S_{ij} \geq I_{j,t-1} - d_{jt} + o_{jt} + x_{jt} - M(1 - q_{ijt}); \forall i, \forall j, \forall t \quad (3.18)$$

Item dengan level persediaan berada pada atau di bawah *can-order* dan *re-order level* bisa dilakukan pemesanan :

$$x_{jt} \leq M(r_{jt} + \sum_{i=1}^I q_{ijt}); \forall j, \forall t \quad (3.19)$$

Jumlah *item* yang *shortage* pada setiap *item* tidak lebih dari permintaan :

$$o_{it} \leq d_{it}; \forall i, \forall t \quad (3.20)$$

Jumlah *item* yang dipesan pada suatu periode tidak melebihi kapasitas penyimpanan:

$$\sum_{i=1}^I x_{it} \leq G; \forall t \quad (3.2)$$

Konstrain variabel biner dan *non-negativity*

$$r_{it}, k_{ijt}, q_{ijt} = \{0,1\}; \forall i, \forall j, \forall t \quad (3.22)$$

$$y_t, I_{it}, x_{it}, o_{it}, s_i, c_{ij}, S_{ij} \geq 0; \forall i, \forall t \quad (3.23)$$

3.5 Tahap Pengembangan Algoritma *Simulated Annealing*

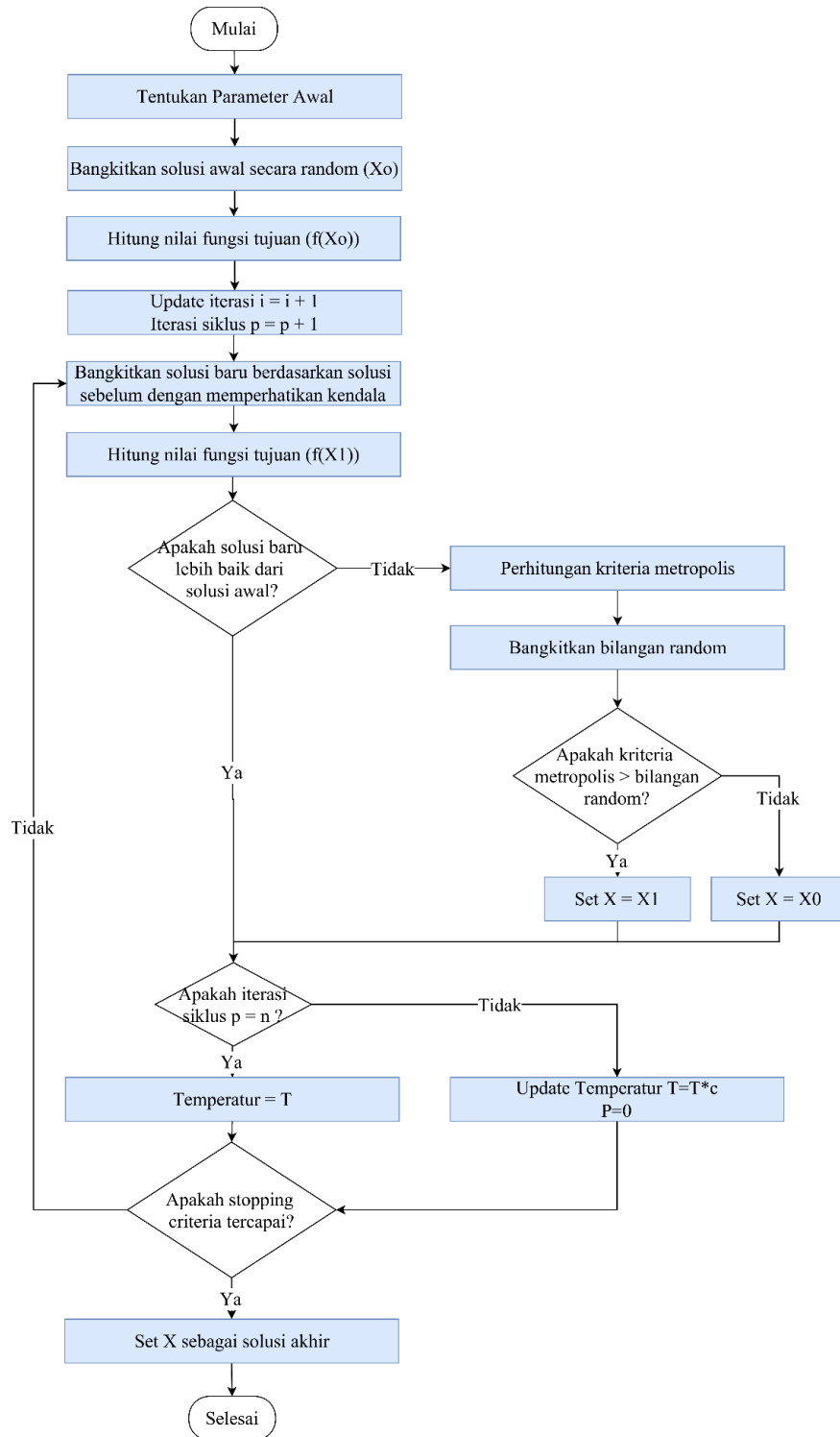
Setelah dibangun model matematis, dilakukan pengembangan algoritma menggunakan penerapan *simulated annealing*. Algoritma SA tersebut akan

digunakan untuk menyelesaikan permasalahan *joint replenishment* menggunakan *multiple can-order level*. Karena adanya fungsi multi tujuan, model akan dicari fungsi *multi objective* fungsi dari metode *global criterion*. Berikut merupakan urutan mencari fungsi multi tujuan:

1. Perhitungan fungsi tujuan 1, didapatkan $Z1^*$.
2. Perhitungan fungsi tujuan 2, didapatkan $Z2^*$.
3. Perhitungan fungsi *multi objective* fungsi tujuan dengan metode global kriteria.

$$\text{Min} \quad F = \left(\frac{Z1^* - Z1}{Z1^*} \right)^2 - \left(\frac{Z2^* - Z2}{Z2^*} \right)^2 \quad (3.24)$$

Tahapan pengerjaan algoritma *simulated annealing* dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Flowchart Penerapan Algoritma Simulated Annealing*

Berikut ini adalah penjelasan dari pengembangan algoritma yang digunakan untuk menyelesaikan permasalahan *joint replenishment* dengan *multiple can-order level*:

1. Menentukan parameter awal

Tahap pertama yang dilakukan algoritma *simulated annealing* adalah penentuan parameter. Parameter yang digunakan dapat mempengaruhi proses pencarian solusi. Parameter-parameter yang terdapat pada algoritma *simulated annealing* antara lain temperatur awal, faktor pereduksi temperatur, siklus perubahan temperature dan iterasi. Temperatur awal dimulai dengan nilai temperature yang tinggi kemudian akan menurun pada setiap iterasi siklusnya. Faktor pereduksi temperatur bernilai 0 sampai 1 yang digunakan sebagai konstanta pereduksi temperatur pada setiap siklus. Siklus perubahan temperatur ditentukan sebagai jumlah iterasi yang dibutuhkan agar temperature dapat turun nilainya. Setiap siklus temperatur ini mengakibatkan menurunnya probabilitas penerimaan untuk solusi yang lebih buruk.

2. Membangkitkan solusi awal secara *random*

Pembangkitan solusi awal dilakukan secara acak bergantung pada *constraint* yang telah ditetapkan. Solusi awal ini merupakan *decision variable* yang dibangkitkan menjadi solusi inisial terbaik untuk sementara, yaitu pada iterasi ke-0. Berikut merupakan gambaran struktur solusi yang dibangkitkan pada permasalahan ini.

Tabel 3.2 Gambaran Struktur Solusi *Re-order Level* (s)

<i>Item</i>	s
1	s ₁
2	s ₂
3	s ₃
...	...
M	s _m

Tabel 3.3 Gambaran Struktur Solusi *Can-order Level* (c)

<i>Item</i>	1	2	3	...	n
1	c ₁₁	c ₂₁	c ₃₁	...	c _{1n}
2	c ₁₂	c ₂₂	c ₃₂	...	c _{2n}
3	c ₁₃	c ₂₃	c ₃₃	...	c _{3n}
...
m	c _{m1}	c _{m2}	c _{m3}	...	c _{mn}

Tabel 3.4 Gambaran Stuktur Solusi *Order-up-to Level* (S)

Item	1	2	3	...	n
1	S ₁₁	S ₂₁	S ₃₁	...	S _{1n}
2	S ₁₂	S ₂₂	S ₃₂	...	S _{2n}
3	S ₁₃	S ₂₃	S ₃₃	...	S _{3n}
...
m	S _{m1}	S _{m2}	S _{m3}	...	S _{mn}

3. Menghitung nilai fungsi tujuan

Perhitungan fungsi tujuan dilakukan dengan memperhitungkan fungsi *multi objective* yang dibangun melalui metode *global criterion* pada persamaan 2.29 dengan mempertimbangkan kendala-kendala yang ada.

4. Memperbarui siklus dan iterasi

Update iterasi dilakukan untuk mengetahui seberapa banyak pengulangan yang dilakukan hingga mendapatkan solusi terbaik. Pengulangan ini juga bisa dilakukan sebagai *stopping criteria*. Sedangkan, *update* siklus dilakukan untuk memberikan waktu pada saat iterasi ke berapa dilakukan penurunan temperatur. Nilai penurunan temperature adalah sebesar factor pereduksi yang telah ditentukan di penentuan parameter awal.

5. Membangkitkan solusi baru

Pembangkitan solusi baru dilakukan dengan perkalian antara bilangan acak dan solusi sebelumnya. Solusi yang dibangkitkan juga dibatasi dengan batas atas dan batas bawah. Berikut merupakan rumus pembangkitan bilangan *random*:

$$S_{i+1} = (S_i + \text{bilanganrandom} * K)$$

dimana,

i = Iterasi

S_i = Nilai *decision variable* pada iterasi ke i

K = Konstanta pengali bilangan *random*

6. Membandingkan nilai fungsi tujuan solusi baru dengan solusi sebelumnya

Setelah didapatkan struktur solusi yang baru, maka dapat dicari nilai fungsi tujuannya. Dari kedua fungsi fungsi *multi objective* tersebut akan dilakukan perbandingan.

- Jika solusi yang baru lebih baik daripada solusi di iterasi sebelumnya, maka solusi yang baru dinyatakan sebagai solusi terbaik untuk iterasi tersebut.
- Jika solusi yang baru lebih buruk daripada solusi sebelumnya, maka solusi tersebut akan masuk pada kriteria metropolis berdasarkan probabilitas *Boltzmann* pada persamaan 2.28. Bilangan *random* akan dibangkitkan untuk selanjutnya dibandingkan nilainya dengan probabilitas *Boltzmann*. Jika bilangan *random* nilainya di atas probabilitas *Boltzmann*, maka solusi yang baru tidak diterima, dan solusi pada iterasi sebelumnya dinyatakan sebagai solusi baru. Sebaliknya, jika bilangan *random* nilainya dibawah atau sama dengan probabilitas *Boltzmann* maka solusi tersebut merupakan solusi *S* yang baru. Hal tersebut digunakan agar algoritma dapat menerima solusi yang lebih buruk untuk mendapatkan solusi yang lebih baik berikutnya atau menghindari terjebaknya *local optima*.

7. Kriteria pemberhentian

Dalam penelitian ini, kriteria pemberhentian yang digunakan adalah nilai fungsi *multi objective* yang sudah tidak berubah secara signifikan dengan variansi yang sangat kecil mencapai *epsilon*, jumlah iterasi maksimum, dan tercapainya nilai temperatur yang sangat kecil. Jika kriteria pemberhentian belum tercapai, maka iterasi dan siklus akan diulang dan diperbarui sampai kondisi yang diinginkan tercapai. Jika *stopping criteria* telah tercapai, maka algoritma yang dibangun akan berhenti dan menerima output berupa waktu komputasi yang dibutuhkan dan nilai dari setiap *decision variable*.

3.6 Validasi Model dan Algoritma

Validasi merupakan proses pengecekan apakah model yang dibuat mampu merepresentasikan model konseptual dari *multi objective multiple can-order level* pada *can-order policy*. Validasi algoritma didapatkan dengan membandingkan antara perhitungan eksak dan hasil dari uji algoritma. Validasi dilakukan dengan data uji berukuran 3 produk, 3 periode, dan 1 *supplier*, serta data uji berukuran 5 produk, 3 periode, dan 2 *supplier*.

3.7 Eksperimen

Eksperimen dilakukan dua kali yakni eksperimen untuk perhitungan eksak dan eksperimen untuk algoritma *simulated annealing*. Eksperimen untuk perhitungan eksak dilakukan dengan menerjemahkan model ke dalam bahasa *software* Lingo sedangkan eksperimen algoritma dilakukan dengan menerjemahkan model sesuai dengan algoritma *simulated annealing* ke dalam bahasa *software* Matlab. Eksperimen dilakukan terhadap set data studi kasus *spare part inventory* PT INALUM (Persero) dengan menguji parameter algoritma terlebih dahulu dari sampel data uji studi kasus.

3.8 Analisis dan Interpretasi Hasil

Analisis dilakukan dengan membandingkan antara hasil eksperimen pada perhitungan eksak dengan algoritma *simulated annealing*. Kriteria perbandingan yang digunakan adalah performansi masing-masing metode yang dilihat dari waktu komputasi dan kualitas solusi yang dihasilkan, selain itu juga dilakukan perbandingan performansi terkait perubahan parameter yang digunakan sebagai input dalam algoritma *simulated annealing*. Hasil dari eksperimen juga akan dianalisa terhadap kondisi riil di perusahaan, untuk mendapatkan berapa efisiensi yang bisa didapatkan dari model yang dibangun.

3.9 Penarikan Kesimpulan dan Saran

Tahap ini merupakan tahapan akhir dalam penelitian ini. Kesimpulan didapatkan berdasarkan hasil analisis dan interpretasi hasil dari eksperimen yang telah dilakukan. Kesimpulan juga menjawab tujuan yang ingin dicapai dalam melakukan penelitian. Setelah penarikan kesimpulan juga dirumuskan saran-saran terkait pengembangan yang mungkin dapat dilakukan pada penelitian selanjutnya.

BAB 4

PENGEMBANGAN MODEL DAN ALGORITMA

Pada bab pengembangan model dan algoritma ini akan dijelaskan tentang tahap-tahap yang dilakukan dalam mengembangkan model *multi objective COP multiple can-order level* menggunakan algoritma *simulated annealing* yang digunakan dalam penyelesaian permasalahan.

4.1 Pengembangan Algoritma *Simulated Annealing*

Pengembangan algoritma didasarkan pada model konseptual yang telah dibuat. Berikut ini adalah penjelasan langkah-langkah yang terdapat dalam algoritma SA yang digunakan :

Langkah 1 : Inisialisasi

Parameter algoritma SA yang digunakan untuk menyelesaikan contoh kasus ini adalah :

- Faktor pereduksi temperatur (cr)
- Jumlah siklus (n)

Parameter temperatur awal ditentukan dengan mengambil nilai rata-rata dari nilai fungsi tujuan dari nilai random solusi yang dibangkitkan. Temperatur awal ditentukan dengan metode ini karena nilai temperatur yang terlalu kecil dibandingkan dengan nilai perubahan fungsi tujuan akan mengakibatkan besar probabilitas boltzman mendekati 0 sehingga komputasi akan terjebak *dalam local optima* pada iterasi awal, sedangkan jika nilai temperatur terlalu besar maka akan dibutuhkan waktu yang cukup lama untuk mencapai konvergensi.

Input parameter yang digunakan pada permasalahan ini adalah besarnya permintaan setiap item pada setiap *planning horizon*, jumlah persediaan pada periode ke-0, serta biaya-biaya untuk menghitung fungsi tujuan meminimasi total *inventory cost*. Data yang akan dilakukan pengujian adalah data yang berasal dari *supplier* A dengan 3 item dan 3 periode. Sedangkan pada proses validasi digunakan factor pereduksi temperature sebesar 0.6 dan jumlah maksimum iterasi siklus sebesar 100. Berikut merupakan contoh data-data yang digunakan untuk pengujian algoritma.

Tabel 4.1 Data Permintaan, Volume, dan Persediaan Awal untuk Model Validasi

Material Number	Vendor	Initial Inventory	Volume (cm ³)	Demand		
				1	2	3
140101000822	B	0	443.7	0	1	0
140101000823	B	1	1841.508	1	0	3
140118000075	B	6	116.28	8	4	4

Tabel 4.2 Data Biaya Persediaan untuk Model Validasi

Material Number	Price	Holding Cost	Ordering Cost	Backordering Cost
140101000822	\$ 323.42	\$ 33.96	\$ 25.70	\$ 371.93
140101000823	\$ 272.13	\$ 28.57	\$ 24.67	\$ 312.95
140118000075	\$ 516.24	\$ 54.21	\$ 29.56	\$ 774.36

Langkah 2: Pembangkitan Solusi

Di dalam algoritma *simulated annealing*, solusi dibangkitkan secara *random*. Solusi awal yang digunakan terdiri dari beberapa variabel seperti, *re-order level* (s_i), *can-order level* (c_{ij}), *order-up-to level* (S_{ij}), dan jumlah item yang shortage (o_{it}). Berikut ini adalah masing-masing cara yang digunakan untuk membangkitkan sampel awal:

$$s = \text{round}(\text{rand} * (s_{\text{upper}} - s_{\text{lower}}) + s_{\text{lower}})$$

Batas bawah s bernilai batas minimum *safety stock* setiap item. Pada proses validasi akan digunakan batas s bernilai 0. Sedangkan batas atas s adalah nilai maksimum interval demand di setiap item.

Tabel 4.3 Nilai *re-order level* (s_i) yang Dibangkitkan secara Random

Item (i)	s_i
1	0
2	1
3	4

Nilai tersebut menunjukkan nilai batas minimum kapan harus dilakukan pemesanan atau yang biasa disebut dengan *reorder level*, untuk setiap itemnya.

$$c = \text{round}(\text{rand} * (c_{\text{upper}} - c_{\text{lower}}) + c_{\text{lower}})$$

Batas bawah c sama dengan nilai s yang telah dibangkitkan. Sedangkan batas atas nilai c adalah nilai maksimum *demand* dari semua periode untuk setiap item.

Tabel 4.4 Nilai *can-order level* (c_{ij}) yang Dibangkitkan secara *Random*

c_{ij}	1	2	3
1	0	3	4
2	0	3	6
3	1	2	8

Nilai *can-order level* menunjukkan bahwa batas minimum inventory item bisa dilakukan pemesanan ketika satu atau lebih item yang terkorrelasi nilainya berada dibawah *re-order level*. Misalnya saja, jika item 2 nilainya dibawah s_2 , maka model akan melihat apakah item 1 berada dibawah c_{21} dan apakah item 3 berada dibawah c_{23} . Jika nilai item yang terpengaruh berada pada atau dibawah *can-order level*, maka pada item tersebut harus dilakukan pesanan sebanyak *order-up-to level* yang bersesuaian. Berikut merupakan nilai *order-up-to level* yang dibangkitkan.

$$S = \text{round}(\text{rand} * (S_{\text{upper}} - S_{\text{lower}}) + S_{\text{lower}})$$

Batas bawah S sama dengan nilai c yang telah dibangkitkan. Sedangkan batas atas nilai S adalah jumlah semua *demand* dalam *planning horizon* untuk setiap item.

Tabel 4.5 Nilai *order-up-to level* (S_{ij}) yang Dibangkitkan secara *Random*

S_{ij}	1	2	3
1	1	4	9
2	1	3	7
3	1	2	8

Nilai S_{ij} berhubungan dengan nilai *can-order level* dan *re-order level*. Jika suatu item nilainya dibawah *re-order level*, item akan dipesan mencapai S_{ii} . Contohnya saja pada item 2 yang nilainya dibawah *re-order level*, maka pemesanan untuk item 2 akan dilakukan mencapai S_{22} . Bila item selain item 2 nilainya tidak dibawah *re-order level*-nya, tetapi berada pada atau dibawah *can-order level* akibat item 2, maka pemesanan akan dilakukan mencapai S_{23} untuk item 3 dan S_{21} untuk item 1.

Dari solusi yang dibangkitkan tersebut, akan didapatkan nilai r dan k . Jika level persediaan jatuh pada atau di bawah *re-order level*, maka r bernilai 1. Jika tidak, maka nilai r sama dengan 0. Pada kondisi r bernilai 1, akan menyebabkan item dipesan pada periode tersebut sejumlah *order-up-to level* S dari item itu sendiri.

Sementara itu, jika level persediaan dari suatu *item* berada pada atau di bawah *can-order level* dari *item* yang berkorelasi, maka k bernilai 1. Selain itu, nilai k bernilai 0. Output dari nilai r dan k akan mempengaruhi besarnya nilai q . Jika r dan k bernilai 0, maka nilai q_{ijt} juga bernilai 0. Selain itu, nilai q bisa bernilai 1 atau 0 dengan hanya ada satu q yang bernilai satu sepanjang item i . Hal ini dilakukan untuk memilih *order-up-to level* pada kondisi apa yang akan digunakan. Pada kondisi q bernilai 1, akan menyebabkan *item* dipesan pada periode tersebut sejumlah *order-up-to level* S dari *item* yang terpicu untuk dipesan. Sehingga untuk mendapatkan nilai q akan diacak menggunakan cara sebagai berikut:

$$q(i, :, t) = (\text{randperm}(\text{Jumlah Item}) = \text{Jumlah Item})$$

Berikut merupakan hasil solusi variabel q yang telah dibangkitkan.

Tabel 4.6 Hasil Solusi Variabel biner q

q_{ijt}		Periode											
		0			1			2			3		
		1	2	3	1	2	3	1	2	3	1	2	3
Item	1	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	1	0	0	0	0	1	0	0

Tabel tersebut menunjukkan bahwa saat q_{321} bernilai 1, maka item akan dipesan pada periode 1 mencapai nilai S_{32} . Kemudian akan dicari besarnya keseimbangan persediaan di setiap periode menggunakan Persamaan 3.3. Jika terdapat level persediaan yang bernilai negatif, akan dibangkitkan solusi kembali secara acak sampai tidak ada level persediaan yang bernilai negatif.

Dari solusi yang sudah dibangkitkan didapatkan output jumlah pesanan x dan *inventory level* I sebagai berikut.

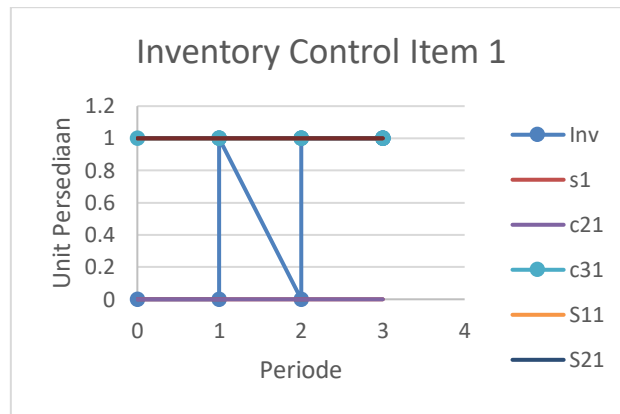
Tabel 4.7 Level Persediaan Hasil Pembangkitan Variabel *Random*

<i>Inventory</i>		Periode			
		0	1	2	3
Item	1	0	1	1	1
	2	1	3	3	3
	3	6	8	8	8

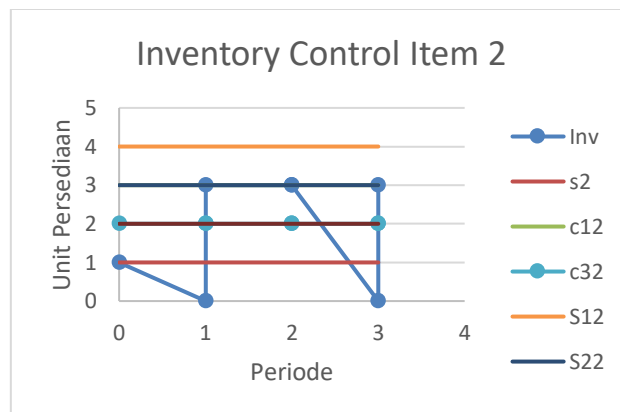
Tabel 4.8 Jumlah Pemesanan Hasil Pembangkitan Variabel *Random*

<i>Order Quantity x</i>		Periode			
		0	1	2	3
Item	1	0	1	1	0
	2	0	3	0	3
	3	0	10	4	4

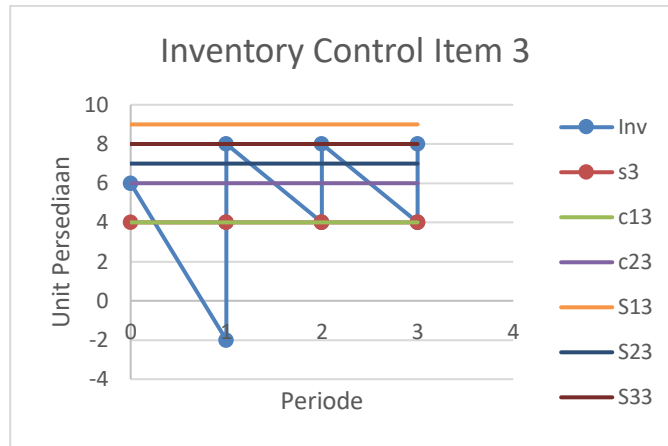
Berikut merupakan gambaran hasil *level* persediaan dan jumlah pemesanan dari solusi yang telah dibangkitkan.



Gambar 4.1 Skema Persediaan Hasil Pembangkitan Bilangan *Random* pada *Item 1*



Gambar 4.2 Skema Persediaan Hasil Pembangkitan Bilangan *Random* pada *Item 2*



Gambar 4.3 Skema Persediaan Hasil Pembangkitan Bilangan Random pada Item 3

Jika kita lihat pada grafik tersebut dan nilai solusi yang telah dibangkitkan, ketiga item tersebut nilainya berada pada dan dibawah *re-order level* setiap item. Sehingga pemesanan setiap item di periode 1 misalnya, akan ditambahkan persediaan mencapai *Sii*. Jika dilihat pada item 2 periode 2, tidak dilakukan pemesanan karena nilainya tidak dibawah *re-order level*, maupun *can-order level* item 2 akibat item 1 dan item 3. Dari output tersebut, akan dihitung besarnya nilai fungsi tujuan yang akan dianalisa.

Langkah 3: Perhitungan Fungsi Tujuan

Perhitungan fungsi tujuan dilakukan setelah didapatkan nilai *decision variable*. Fungsi tujuan 1 dihitung berdasarkan model matematis yang telah dibangun pada Persamaan 3.1, atau jika dirumuskan ke dalam Matlab adalah sebagai berikut:

$$Z1 = \text{sum}(\text{ceil}(\text{sum}(x.*\text{repmat}(v,1,T))/\text{cap})))$$

Fungsi tujuan 2 dihitung berdasarkan model matematis yang telah dibangun pada Persamaan 3.2. Sebelum menghitung total biaya, terlebih dahulu dilakukan perhitungan *major ordering cost*. Jika total biaya pembelian *item* dalam satu periode dan *supplier* yang sama di bawah \$769.23 maka dikenakan *major ordering cost* sebesar \$38.46. Jika total biaya pembelian *item* dalam satu periode dan *supplier* yang sama di atas \$769.23 maka dikenakan *major ordering cost* sebesar \$153.85.

Apabila total biaya pembelian berada diantara \$769.23 dan \$7692.3, maka dikenakan biaya sebesar \$76.92.

Jika dirumuskan ke dalam Matlab, perhitungan total biaya adalah sebagai berikut:

```
Z2 = sum(sum(x.*repmat(price,1,T)+r.*repmat(CO,1,T)+
    o.*repmat(CB,1,T)))+sum(sum(Inv(:,2:T)
    .*repmat(CH,1,T-1)));
```

Pada fungsi tujuan 1 didapatkan jumlah *carrier* sebesar 2, sedangkan nilai fungsi tujuan 2 didapatkan total biaya persediaan sebesar \$ 14463.

Pada penyelesaian permasalahan multi tujuan, model akan menyelesaikan kedua *problem* fungsi tujuan 1 dan fungsi tujuan 2 terlebih dahulu. Kemudian, model dilakukan running kembali dengan merubah fungsi multi tujuan seperti rumus dibawah ini :

$$fglobal = ((fbest1 - Z1) / fbest1)^2 + ((fbest2 - Z2) / fbest2)^2$$

Z1 dan Z2 merupakan fungsi tujuan yang akan dicari, sedangkan fbest1 dan fbest2 merupakan nilai optimal dari masing-masing fungsi tujuan yang sudah didapatkan sebelumnya.

Langkah 4: Pembangkitan Solusi Baru

Dari solusi acak yang telah dibangkitkan, akan diulangi sebanyak 100 kali, kemudian dicari nilai fungsi tujuan yang paling baik sebagai *initial solution*. Sementara itu, nilai rata-rata dari 100 data tersebut akan digunakan sebagai nilai temperature awal. *Initial solution* yang telah dibangkitkan tersebut akan diubah menjadi nilai terdekatnya atau biasa disebut *neighborhood search* sebagai solusi baru.

Perubahan solusi yang telah dibangkitkan dilakukan dengan cara melakukan kombinasi perubahan penambahan 1 satuan, pengurangan 1 satuan, atau tidak dilakukan perubahan (bertambah 0 satuan) pada variabel s , c , S , dan o . Misalnya pada variabel s_3 yang bernilai 2 saat dibangkitkan solusi yang baru, ada kemungkinan nilainya berubah menjadi 1, tetap bernilai 2, maupun bertambah 1 menjadi 3. Perubahan tersebut dibatasi dengan nilai minimumnya tidak kurang dari

0 dan tidak melebihi batas maksimum tiap variabel. Sehingga, dapat dibuat rumus seperti berikut:

```
snew(i)      =min(max(s(i)+floor(3*rand()-1),0),supper)
cnew(i,j)    =min(max(c(i,j)+floor(3*rand()-1),snew(i)),cupper);
Snew(i,j)    =min(max(S(i,j)+floor(3*rand()-1),cnew(i,j)),Supper);
onew(i,t+1)  =min(max(o(i,t+1)+floor(3*rand()-1),0),
                floor(Max_%Stockout*demand(i,t+1)));
```

Dari pembangkitkan solusi baru tersebut akan didapatkan output nilai variabel lain seperti pada langkah 2, kemudian didapatkan nilai fungsi tujuan yang baru. Fungsi tujuan 1 bernilai 2, sedangkan fungsi tujuan 2 berubah menjadi 15541.

Langkah 5: Membandingkan Solusi Lama dengan Solusi Baru

Setelah mendapatkan solusi baru kemudian dilakukan perhitungan Δf yang merupakan selisih antara fungsi tujuan yang didapatkan dengan solusi baru dan solusi yang lama. Jika nilai Δf *negative* maka solusi yang baru lebih baik dibandingkan dengan solusi yang sebelumnya sehingga solusi baru akan diterima. Tetapi, jika solusi baru tidak lebih baik dibandingkan dengan solusi yang lama, maka akan dilakukan perhitungan kriteria metropolis untuk menentukan apakah solusi yang baru diterima atau tidak. Perhitungan kriteria metropolis dilakukan dengan membandingkan antara probabilitas boltzman dengan bilangan random, jika bilangan random lebih kecil dari kriteria metropolis maka solusi yang baru akan diterima jika tidak maka solusi yang baru akan ditolak.

$$\text{deltaf} = Z2 - f_{\text{best}2}$$

$$\text{deltaf} = 1078$$

Dalam perhitungan Δf diatas, didapat nilai Δf positif, maka dapat dilakukan perhitungan kriteria metropolis untuk mengetahui probabilitas penerimaan:

$$\text{Temperatur} = 15798$$

$$P(E) = e^{-AE/kT}$$

$$P(E) = e^{1078/15798}$$

$$P(E) = 0.934$$

Selanjutnya digenerate bilangan random yang didapatkan sebesar 0.326, karena nilai bilangan random kurang dari probabilitas boltzman maka dapat

disimpulkan solusi yang baru diterima sehingga dapat ditetapkan nilai solusi saat ini sama dengan solusi yang baru dan komputasi dilakukan hingga mencapai *stopping criteria*-nya.

Langkah 5: *Update Iterasi, Siklus dan Temperatur*

Update iterasi dan siklus dilakukan setelah didapatkan solusi yang baru. Sedangkan *update* nilai temperatur dilakukan setelah tercapai jumlah siklus sebesar n yang merupakan nilai maksimal jumlah siklus. Setelah mencapai jumlah siklus n , temperatur akan direduksi dengan menggunakan faktor pereduksi temperatur (cr). Selain itu, setelah mencapai jumlah siklus n , nilai siklus kembali ditetapkan menjadi sama dengan satu.

Langkah 6: *Stopping Criteria*

Penetapan *stopping criteria* akan menentukan kapan algoritma berhenti melakukan komputasi. Dalam penelitian ini *stopping criteria* yang digunakan adalah jumlah iterasi maksimum. Jika *stopping criteria* belum tercapai, maka iterasi akan diulang dengan meng-*update* iterasi dan siklus serta tempaeratur. Sementara itu, jika *stopping criteria* telah tercapai maka akan ditampilkan *outputan* dari algoritma *simuated annealing* yang telah dibuat yakni menampilkan nilai fungsi tujuan, nilai variabel keputusan, serta waktu komputasi.

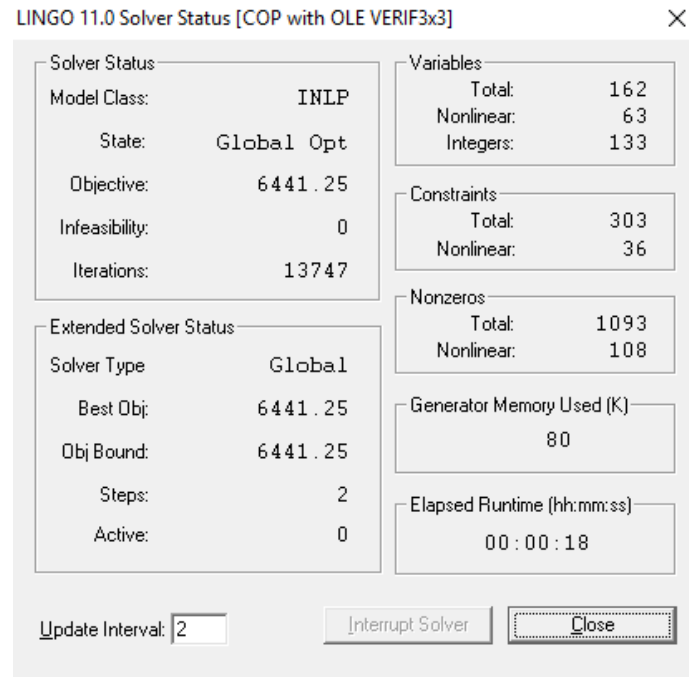
4.2 Validasi dan Verifikasi

Validasi merupakan tahapan yang dilakukan untuk mengetahui apakah model yang dibuat telah mampu merepresentasikan permasalahan yang diselesaikan. Validasi dalam penelitian dilakukan untuk model matematis dan juga untuk algortma *simulated annealing*.

4.2.1 Validasi dan Verifikasi Model Matematis

Verifikasi model eksak dilakukan dengan melakuakan evaluasi struktur model yang digenerate dalam *software* LINGO. Evaluasi model didasarkan pada apakah model yang digenerate telah memiliki struktur yang sesuai dengan model matematisnya. Validasi model matematis dilakukan dengan membandingkan antara

logika perhitungan hasil komputasi dengan logika perhitungan manualnya. Berikut merupakan hasil perhitungan fungsi tujuan 1 menggunakan *global solver* dengan metode *branch and bound*.



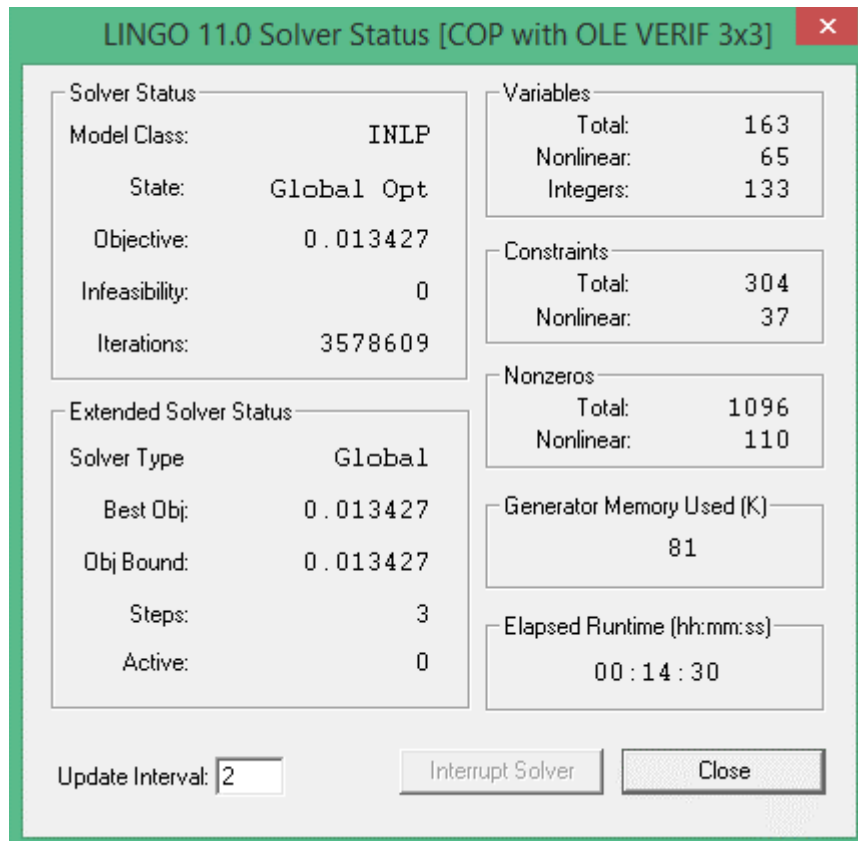
Gambar 4.4 Hasil Perhitungan Fungsi Tujuan 1 dengan Metode Eksak pada Data Verifikasi

Pada perhitungan tersebut didapatkan total biaya minimum sebesar \$6441.25 dan jumlah *carrier* sebesar 3 unit. Kemudian juga dilakukan perhitungan pada fungsi tujuan 2 dengan hasil sebagai berikut.



Gambar 4.5 Hasil Perhitungan Fungsi Tujuan 2 dengan Metode Eksak pada Data Verifikasi

Pada perhitungan fungsi tujuan 2 didapatkan total biaya minimum sebesar \$7919.6 dan jumlah *carrier* sebesar 1 unit. Setelah didapatkan nilai *global optimum* dari setiap fungsi tujuan, dilakukan *running model* kembali dengan fungsi multi tujuan menggunakan global kriteria. Berikut merupakan hasil dari perhitungan eksak dengan fungsi multi obyektif.



Gambar 4.6 Hasil Perhitungan Fungsi Multi Tujuan dengan Metode Eksak pada Data Verifikasi

Hasil metode eksak menggunakan fungsi obyektif, didapatkan nilai *multi objective* sebesar 0.0134, dengan fungsi tujuan 1 sebesar 1 unit dan fungsi tujuan 2 sebesar \$7187.629. Berikut merupakan hasil solusi dari perhitungan eksak.

Tabel 4.9 *Re-order Level* (s) Hasil Perhitungan Metode Eksak

Item	s
1	0
2	0
3	0

Tabel 4.10 *Can-order Level* (c) Hasil Perhitungan Metode Eksak

c		Item		
		1	2	3
Item	1	0	0	0
	2	0	0	0
	3	0	0	0

Tabel 4.11 *Order-Up-To Level (S)* Hasil Perhitungan Metode Eksak

S		Item		
		1	2	3
Item	1	0	0	0
	2	0	0	0
	3	1	3	8

Tabel 4.12 Jumlah Pesanan Hasil Perhitungan Metode Eksak

Order Quantity		Periode			
		1	2	3	4
Item	1	0	1	0	0
	2	0	3	0	0
	3	0	10	0	0

Tabel 4.13 *Level Persediaan* Hasil Perhitungan Metode Eksak

Inventory		Periode			
		1	2	3	4
Item	1	0	1	0	0
	2	1	3	3	0
	3	6	8	4	0

Tabel 4.14 Jumlah *Item Shortage* Hasil Perhitungan Metode Eksak

Shortage		Periode			
		1	2	3	4
Item	1	0	0	0	0
	2	0	0	0	0
	3	0	0	0	0

Jika dilihat pada hasil, tidak ada dari fungsi tersebut yang melanggar kendala dari model yang ditentukan. Hasil dari fungsi multi obyektif juga tidak berselisih jauh dari fungsi global optimal setiap fungsi tujuan. Sehingga dapat dikatakan, model matematis yang telah dibangun telah terverifikasi dan valid.

4.2.2 Validasi dan Verifikasi Algoritma

Verifikasi algoritma dilakukan mengetahui apakah terdapat *error* dalam melakukan komputasi dengan *software* MATLAB. Langkah pertama adalah mencoba *running* model untuk menyelesaikan fungsi tujuan 1, seperti berikut ini :

```
>> [sbest, cbest, Sbest, obest, Invbest, xbest, rbest, kbest,
qbest,waktukomputasi,fbest1]=SACOLZ1(demand,Inv0,price,CH,CO,CB,v,
cap,100,0.6)
```

```
sbest =
```

```
1
0
4
```

```
cbest =
```

```
1      1      2
1      3      3
8      6      5
```

```
Sbest =
```

```
1      1      1
2      4      4
11     16     14
```

```
waktukomputasi =
```

```
19.844
```

```
fbest1 =
```

```
1
```

Kemudian juga dicoba *running* model untuk menyelesaikan fungsi tujuan 2, seperti berikut ini :

```
>> [sbest, cbest, Sbest, obest, Invbest, xbest, rbest, kbest,
qbest,waktukomputasi,fbest2]=SACOLZ2(demand,Inv0,price,CH,CO,CB,v,
cap,100,0.6)
```

```
sbest =
```

```
0
0
0
```

```
cbest =
```

```
0      0      0
0      0      0
```

```

0      0      0

```

```

Sbest =

```

```

0      1      1
2      0      4
6     11      0

```

```

waktukomputasi =

```

```

23.594

```

```

fbest2 =

```

```

6.4412e+03

```

Dari hasil penyelesaian tiap individu fungsi tujuan, hasil model algoritma SA dan penyelesaian secara eksak memiliki nilai solusi yang sama yaitu pada fungsi tujuan 1 sebesar 1 *carrier* dan fungsi tujuan 2 sebesar \$6441.2. Hasil tersebut sama dengan hasil pada perhitungan eksak.

Setelah setiap model single obyektif telah valid, akan dilanjutkan dengan menggunakan algoritma untuk menyelesaikan permasalahan multi obyektif dengan global kriteria. Berikut merupakan hasil *running* model multi obyektif.

```

>> [sbest, cbest, Sbest, obest, Invbest, xbest, rbest, kbest,
qbest,waktukomputasi,Z1opt,Z2opt,fbest]=SACOLMULTI(demand,Inv0,pri
ce,CH,CO,CB,v,cap,100,0.6)

```

```

sbest =

```

```

0
0
0

```

```

cbest =

```

```

0      0      0
0      0      0
0      0      0

```

```

Sbest =

```

1	2	8
1	4	8
1	3	8

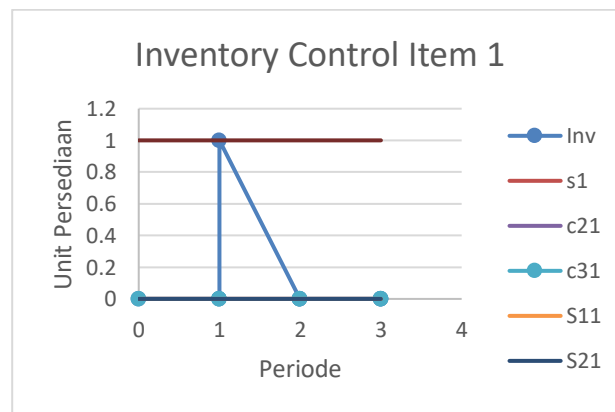
waktukomputasi =
1.5416e+02

z1opt =
1

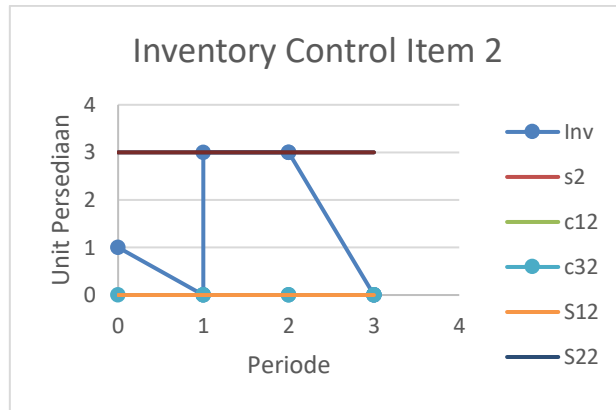
z2opt =
7.1870e+03

fbest =
0.0130

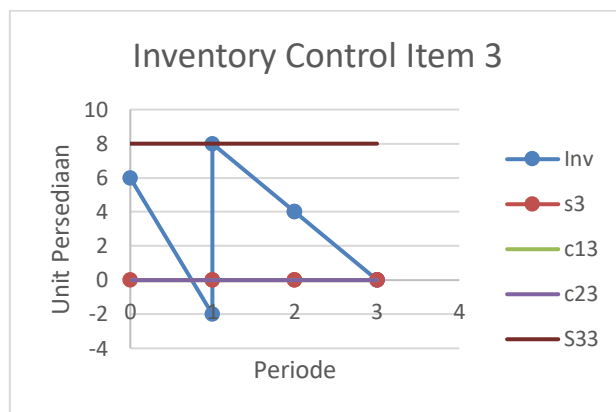
Setelah didapatkan hasilnya, dapat diketahui grafik persediaan yang dihasilkan, seperti pada gambar berikut ini.



Gambar 4.7 Skema Persediaan Hasil Verifikasi Algoritma SA pada Item 1



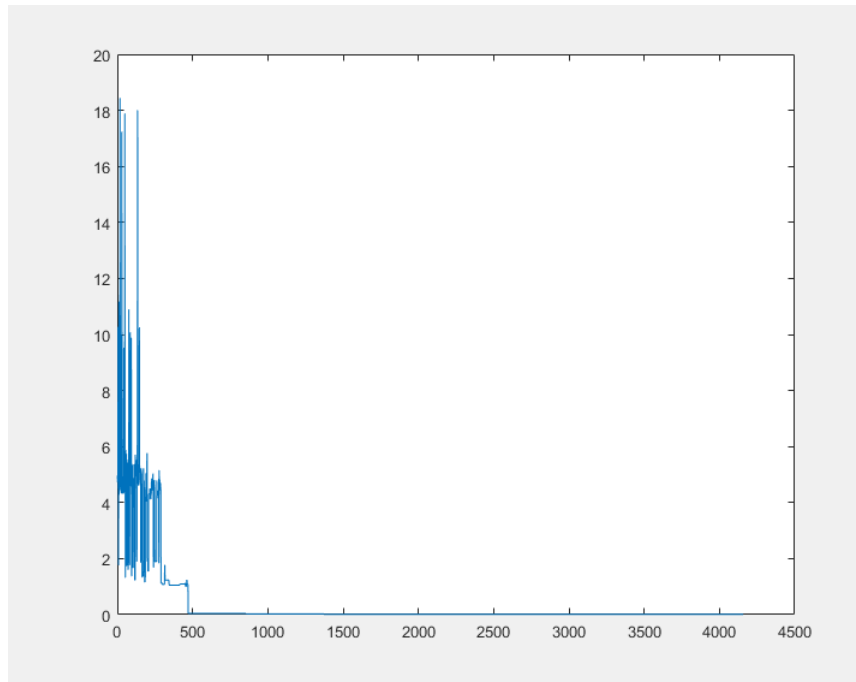
Gambar 4.8 Skema Persediaan Hasil Verifikasi Algoritma SA pada Item 2



Gambar 4.9 Skema Persediaan Hasil Verifikasi Algoritma SA pada Item 3

Dari grafik persediaan tersebut dapat dilihat bahwa terjadi perubahan antara bilangan random yang telah dibangkitkan dan hasil algoritma. Pada periode 1, item 3 nilainya berada dibawah *re-order level*, sedangkan item 1 dan 2 nilainya tepat berada di *can-order level* akibat dari item 3. Sehingga item 3 akan dilakukan pemesanan mencapai S_{33} , item 1 dilakukan pemesanan mencapai S_{31} dan item 2 dilakukan pemesanan mencapai S_{32} . Hasil tersebut menunjukkan bahwa semua pesanan item yang memiliki kesamaan *supplier* tersebut dilakukan pada periode yang sama untuk meminimasi jumlah *carrier* yang mendapatkan hasil 1 unit *carrier*.

Berikut merupakan plot grafik bagaimana algoritma *Simulated Annealing* menemukan nilai fungsi tujuan di setiap iterasinya.



Gambar 4.10 Plot Grafik Nilai Fungsi Multi Tujuan pada Algoritma SA

Grafik di atas menunjukkan bahwa algoritma SA yang dibangun akan menuju solusi yang lebih baik semakin banyak iterasinya. Naik turunnya fungsi tujuan di awal-awal iterasi menunjukkan bahwa terjadi penerimaan solusi yang lebih buruk dari solusi yang telah dibangkitkan. Semakin banyak iterasi semakin menurun temperatur, sehingga peluang solusi yang lebih buruk diterima semakin tinggi.

Dari hasil perhitungan model multi obyektif, hasilnya telah sama dengan perhitungan eksak. Tetapi, untuk mencapai hasil yang sama dibutuhkan replikasi dengan jumlah yang besar yang akan ditunjukkan pada subbab 5.2. Berikut merupakan rekapitulasi hasil verifikasi dan validasi menggunakan data 3 item dan 3 periode.

Tabel 4.15 Perbandingan Hasil Verifikasi Algoritma SA dengan Perhitungan Eksak

Fungsi Tujuan		Nilai Solusi			Waktu Komputasi (detik)		
		Eksak	SA	GAP%	Eksak	SA	GAP%
Single Obyektif	Fungsi Tujuan 1	1	1	0.00%	3	19.8	560.00%
	Fungsi Tujuan 2	6441.25	6441.25	0.00%	18	23.6	31.11%
Multi Obyektif	Fungsi Tujuan 1	1	1	0.00%	870	40.8	-95.31%
	Fungsi Tujuan 2	7187.63	7187.63	0.00%	870	40.8	-95.31%

Jika dilihat pada nilai solusi, perhitungan SA tidak memiliki perbedaan yang signifikan pada perhitungan eksak. Perbedaan terjadi saat dilakukan *running* model multi tujuan. Hal ini terjadi karena pada SA dibangkitkan bilangan random yang kemudian akan diperbaiki dengan mencari variabel keputusan berdasarkan *Neighborhood Search*. Sedangkan, jika dilihat pada waktu komputasi, Algoritma SA pada model single obyektif memiliki waktu yang lebih lama dari pada perhitungan eksak. Tetapi, bila dibandingkan dengan model multi obyektif, Algoritma SA lebih cepat dari pada perhitungan eksak. Hal ini dikarenakan terdapat fungsi *non-linear* pada fungsi *multi objective* multi tujuan. Sehingga jika menggunakan perhitungan eksak dibutuhkan waktu yang sangat lama. Sedangkan pada SA, waktu komputasinya relatif stabil bergantung pada jumlah maksimum iterasi dan konstanta pereduksi temperatur (cr).

Setelah model dengan 1 *supplier* dengan 3 item dan 3 periode telah valid, model akan dicoba *running* kembali menggunakan permasalahan *multi supplier* dengan ukuran 2 *supplier* dengan 3 item dari *supplier 1* dan 2 item dari *supplier 2* dengan 3 periode. Berikut merupakan rekapan hasil pada permasalahan kedua yang akan dijelaskan lebih rinci pada subbab 5.3.

Tabel 4.16 Perbandingan Hasil Verifikasi Algoritma SA dengan Perhitungan Eksak
Multi Supplier

Fungsi Tujuan		Nilai Solusi			Waktu Komputasi (detik)		
		Eksak	SA	GAP%	Eksak	SA	GAP%
Single Objektif	Fungsi Tujuan 1	1	1	0.00%	4	36.14	803.50%
	Fungsi Tujuan 2	8788.61	8788.61	0.00%	49	38.15	-22.14%
Multi Objektif	Fungsi Tujuan 1	1	1	0.00%	445	273	-38.65%
	Fungsi Tujuan 2	9644.32	10371	7.53%	445	273	-38.65%

Pada permasalahan multi supplier, didapatkan hasil solusi dengan perbedaan terbesar pada model multi obyektif fungsi tujuan biaya, yaitu sebesar 7.53%. Tetapi jika dilakukan *running* hingga beberapa replikasi, hasil algoritma SA dapat mencapai nilai global optimalnya. Sedangkan pada waktu komputasi hasilnya juga sama bila dibandingkan dengan permasalahan 1 *supplier*, yaitu waktu komputasi SA lebih cepat dibandingkan menggunakan perhitungan eksak.

Dari hasil tersebut, algoritma *Simulated Annealing* dapat dikatakan telah valid, mengacu pada hasil solusinya. Dengan demikian, model yang telah dibangun dapat dilanjutkan untuk permasalahan dengan ukuran yang lebih besar atau pada permasalahan nyata.

BAB 5

EKSPERIMEN DAN ANALISIS

Pada bab eksperimen dan analisis ini akan dijelaskan mengenai eksperimen untuk menguji parameter algoritma dan hasil pengolahan data studi kasus. Selain itu, juga dilakukan analisis perbandingan dari hasil algoritma dan kondisi eksisting.

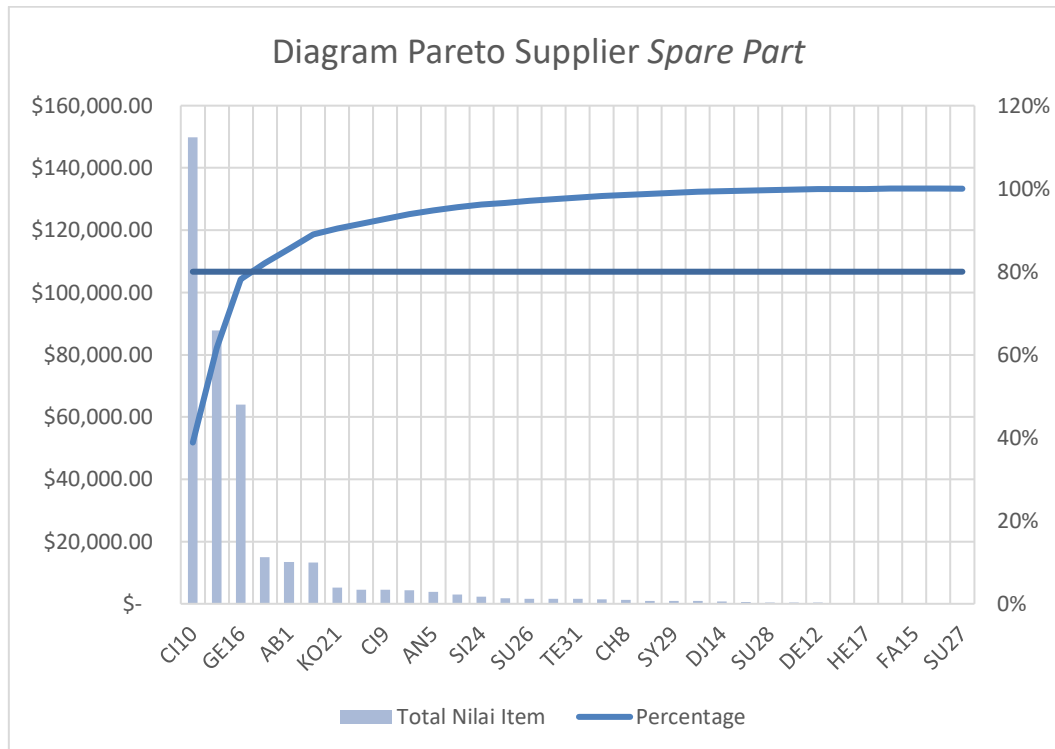
5.1 Pengumpulan Data

PT INALUM (Persero) memiliki kurang lebih 14.000 jenis *spare part*, antara lain *mechanical part*, *electrical part*, fabrikasi lokal, material mekanik, *general goods*, *tools* dan pelumas. *Mechanical part* sendiri merupakan *item* dengan pembelian terbesar yaitu 60% dari pembelian yang dilakukan perusahaan. Proses pengadaan *spare part* tersebut dilakukan melalui sumber beberapa *supplier*. Setiap *supplier* dapat menyuplai lebih dari 1 *item*. Dari 645 *item* pada *forecast* pengadaan *mechanical part* tahun 2018, terdapat 11 *supplier*. Berikut merupakan data *supplier* dan jumlah *item* yang disuplai oleh *supplier* tersebut.

Tabel 5.1 Data *Supplier* dan Jumlah *Item* yang Disuplai

Supplier	Jumlah Item	Total Nilai Item	Supplier	Jumlah Item	Total Nilai Item
CI10	43	\$149,802.58	SU25	7	\$1,429.89
AD2	52	\$87,793.03	CH8	1	\$1,225.75
GE16	62	\$63,927.21	KA19	6	\$904.37
AL4	30	\$15,005.53	SY29	11	\$882.36
AB1	9	\$13,364.06	TA30	2	\$833.07
BU7	12	\$13,296.73	DJ14	20	\$771.73
KO21	12	\$5,248.39	MO23	9	\$605.99
KE20	9	\$4,502.38	SU28	1	\$440.80
CI9	23	\$4,497.10	WO33	1	\$349.31
DA11	8	\$4,320.01	DE12	1	\$338.24
AN5	14	\$3,765.35	BA6	3	\$284.51
AD3	1	\$3,005.63	HE17	1	\$144.05
SI24	2	\$2,292.67	DI13	5	\$54.45
IN18	2	\$1,809.42	FA15	2	\$34.43
SU26	2	\$1,607.54	WI32	1	\$27.92
KO22	12	\$1,539.18	SU27	1	\$11.96
TE31	2	\$1,534.13			

Dari data *supplier* tersebut hanya akan dipilih beberapa *supplier* berdasarkan besarnya nilai item yang disuplai *supplier* tersebut berdasarkan diagram pareto. Jumlah *supplier* yang dipilih didasarkan pada 80% dari total nilai item seperti pada grafik berikut.



Gambar 5.1 Diagram Pareto *Supplier* pada *Spare Part*

Hasil diagram pareto tersebut didapatkan bahwa penyelesaian permasalahan di PT INALUM (Persero) akan diselesaikan dengan data item dari 3 *supplier*, yaitu CI10, AD2, dan GE16 dengan total 157 item.

Data yang akan digunakan di dalam model adalah data *forecast* tahun 2018. Data tersebut ditampilkan pada Lampiran A. Berikut merupakan contoh tabel permintaan *spare part* pada 10 item, yang juga digunakan pada data uji parameter algoritma.

Tabel 5.2 Data Peramalan Permintaan *Spare Part* Tahun 2018

Material Number	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	2018 Total
A2276	0	0	6	2	4	0	0	2	0	0	0	0	14
A4311	76	14	14	14	14	14	16	0	0	0	0	0	162
A0161	0	2	2	1	10	0	0	0	0	0	0	0	15
C0823	1	0	3	3	0	0	0	0	0	1	0	0	8
C1243	0	0	0	0	1	1	2	0	0	0	1	0	5
C0075	8	4	4	12	0	4	8	0	0	0	4	0	44
G0288	0	0	0	3	7	0	3	5	3	2	2	0	25
G1043	14	7	7	21	3	3	6	0	0	0	3	0	64
G0100	0	0	0	3	7	0	3	5	3	2	2	0	25
G0089	7	0	7	14	7	0	0	0	0	0	0	0	35

Data-data lain yang dibutuhkan pada permasalahan pemesanan *spare part* adalah data harga, *initial inventory*, volume, *safety stock*, klasifikasi kritis item, *supplier* tiap item, dan biaya-biaya di dalam persediaan seperti *holding cost* dan *ordering cost*. Berikut merupakan data-data yang dibutuhkan pada penyelesaian model.

Tabel 5.3 Data Atribut *Spare Part*

Material Number	Kode Vendor	Critical Code	Volume (cm ³)	Initial Inventory	Safety Stock	Harga (\$)	Satuan
A2276	AD2	B	141354.18	2	5	134.29	each
A4311	AD2	A	958.3	2	13	69.93	each
A0161	AD2	C	360	6	6	19.78	set
C0823	CI10	C	408.32	1	2	272.13	each
C1243	CI10	A	270000	0	3	2716.83	each
C0075	CI10	A	1162.8	6	16	516.24	each
G0288	GE16	B	5877.23	1	5	72.92	each
G1043	GE16	A	1176	1	13	40.8	each
G0100	GE16	A	544.5	1	6	146.17	each
G0089	GE16	C	56000	7	5	13.02	each

Data biaya *item* meliputi biaya pemesanan, biaya penyimpanan, dan biaya *shortage*. Biaya pemesanan yang diterapkan perusahaan dibedakan menjadi dua, yaitu biaya tetap pemesanan *minor* dan pemesanan *major*. Biaya pemesanan *minor* merupakan biaya yang dikenakan pada masing – masing item yaitu sebesar 2% dari

harga item. Sedangkan, biaya pemesanan *major* merupakan biaya tetap yang dikeluarkan untuk sekali pemesanan. Biaya pemesanan *major* yang ditetapkan perusahaan adalah sebagai berikut.

Tabel 5.4 Biaya Pemesanan *Major Spare Part*

Total Pembelian	Biaya Pemesanan <i>Major</i>
< \$769.23	\$38.46
\$769.23 - \$7692.3	\$76.92
> \$7692.3	\$153.85

Biaya *backorder* akan dikenai bila pada suatu periode terdapat *stockout* dan harus dipesan pada periode tersebut dengan waktu yang lebih cepat. Perusahaan menetapkan ada tambahan *penalty* biaya sebesar 15% untuk kelas item C, 30% untuk item B, dan 50% untuk kelas item A dari biaya pemesanan terhadap item tersebut.

Sedangkan pada biaya penyimpanan ditentukan berdasarkan fraksi bunga dari harga item yang dibeli. Pada PT INALUM (Persero), biaya penyimpanan ditentukan sebesar 10% dari harga *spare part* tiap periodenya.

Untuk mengantarkan *item* ke pabrik, tipe kendaraan setiap *supplier* diasumsikan sama yaitu menggunakan truk L300 berdimensi 6 CBM (144 cm x 30 cm x 242 cm) dengan kapasitas maksimum 1.045.440 cm³.

5.2 Kondisi Eksisting

PT INALUM (Persero) saat ini menerapkan kebijakan persediaan (*s*, *T*) dengan *s* merupakan *re-order level* dan *T* adalah *replenishment period*. Saat persediaan menyentuh atau dibawah *re-order level*, yang ditetapkan perusahaan sebesar *safety stock*, perusahaan akan melakukan pemesanan item sebanyak permintaan selama 3 bulan kedepan dari periode dimana *level* persediaan dibawah *re-order level*. Dari 157 *item* yang disuplai 3 *supplier*, skenario pemesanan yang dilakukan perusahaan dapat dilihat pada Lampiran E.

Dari waktu pemesanan, jumlah pemesanan dan *level* persediaan tersebut dapat dihitung total biaya dan *carrier*. Berikut ini adalah hasil perhitungan biaya persediaan dan total *carrier supplier* pada kondisi eksisting.

Tabel 5.5 Hasil Perhitungan Biaya Persediaan Kondisi Eksisting

Biaya	Nilai
Biaya Pemesanan	\$ 20,756.84
Biaya Pembelian	\$ 406,771.11
Biaya Penyimpanan	\$ 158,657.64
Total Biaya	\$ 586,185.59

Tabel 5.6 Hasil Perhitungan Total *Carrier* Kondisi Eksisting

Supplier	Carrier (unit)
CI10	13
AD2	41
GE16	22
Total	76

Dari hasil perhitungan, total biaya yang dikeluarkan perusahaan sebesar \$586,185.59 dan total jumlah *carrier supplier* sebesar 76 unit. Berikut merupakan rekapitan jumlah pemesanan *item* di tiap perioded dan *supplier*.

Tabel 5.7 Jumlah Pesanan *Item* pada Kondisi Eksisting

Jumlah Item yang Dipesan		Periode												Total
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
Supplier	CI10	204	118	48	207	18	0	174	0	20	50	16	0	855
	AD2	572	0	16	196	0	26	399	0	63	8	0	0	1280
	GE16	1293	0	3	228	121	58	230	0	27	34	0	3	1997
Total		2069	118	67	631	139	84	803	0	110	92	16	3	4132

Hasil perhitungan total biaya dan total *carrier* tersebut akan dibandingkan pada hasil perbaikan menggunakan optimasi *can-order policy* menggunakan algoritma SA.

5.3 Eksperimen Uji Parameter Algoritma SA

Eksperimen dilakukan dengan menggunakan *software* MATLAB dan LINGO 11. *Software* MATLAB digunakan untuk menyelesaikan permasalahan dengan menggunakan algoritma *simulated annealing*. Spesifikasi komputer yang digunakan adalah Intel ® Core™ i5-4570 3.20 G.Hz (CPU), RAM 8 GB.

Dalam algoritma SA terdapat beberapa parameter yang digunakan untuk mengenerate solusi diantaranya adalah *cr* (faktor pereduksi temperatur) dan *n* (banyaknya siklus). Nilai parameter-parameter ini akan mempengaruhi kualitas dan kecepatan dalam men-*generate* solusi sehingga perlu dilakukan uji parameter untuk menentukan nilai parameter yang tepat untuk mendapatkan solusi yang terbaik. Nilai *cr* yang diuji adalah 0.9 dan 0,6. Sedangkan nilai maksimum iterasi tiap siklus yang diuji adalah 100 dan 1000.

5.3.1 Deskripsi Data Uji

Untuk melakukan pengujian parameter algoritma *simulated annealing*, data yang digunakan setiap data uji akan digunakan dengan jumlah *item*, *periode*, dan *supplier* yang berbeda. Berikut merupakan ukuran data yang akan digunakan dalam eksperimen uji parameter.

Tabel 5.8 Data Uji

Data	Produk	Periode	Supplier
Data Uji 1	3	3	1
Data Uji 2	5	3	2
Data Uji 3	10	12	3
Data Studi Kasus	157	12	3

Data - data tersebut diambil dari data studi kasus *spare part inventory* PT INALUM (Persero) melalui sampel dengan memperhitungkan item dan *supplier* yang paling berkontribusi atau memiliki nilai yang tertinggi dari keseluruhan proses pengendalian persediaan. Untuk lebih lengkapnya, data tersebut dapat dilihat pada Lampiran.

5.3.2 Eksperimen Data Uji 1

Berikut merupakan hasil yang diperoleh dari hasil *running* model menggunakan data uji 1 yang telah ditampilkan pada Tabel 4.1 dan Tabel 4.2. Data uji 1 juga digunakan di dalam proses verifikasi dan validasi.

Tabel 5.9 Hasil Uji Parameter Data Uji 1

Cr	Maxit	Replikasi	Jumlah <i>Carrier</i> (unit)	Total Biaya (\$)	Fungsi <i>Multi</i> <i>objective</i>	Waktu Komputasi (detik)
0.6	100	1	3	6441.3	4.000	4.1
		2	1	7919.6	0.053	3.9
		3	1	7919.6	0.053	3.9
		4	2	7745.5	1.041	4
		5	1	7187.6	0.013	4.7
		Rata-rata	1.6	7442.72	1.032	4.12
0.9	100	1	1	7919.6	0.053	19.8
		2	1	7919.6	0.053	23.6
		3	1	7187.63	0.013	19.6
		4	3	6441.3	4.000	21.4
		5	1	7919.6	0.053	18.4
		Rata-rata	1.4	7477.546	0.834	20.56
0.6	1000	1	1	7919.6	0.053	40.7
		2	1	7919.6	0.053	40.1
		3	1	7187.63	0.013	41.7
		4	1	7187.63	0.013	40.7
		5	1	7187.63	0.013	40.8
		Rata-rata	1	7480.418	0.029	40.8
0.9	1000	1	1	7187.63	0.013	191.1
		2	1	7919.6	0.053	189.2
		3	1	7187.63	0.013	190.3
		4	1	7187.63	0.013	191
		5	1	7919.6	0.053	194
		Rata-rata	1	7480.418	0.029	191.12

Dari uji parameter data uji 1 didapatkan rata-rata nilai objektif terbaik didapatkan dengan menggunakan parameter maksimum iterasi siklus sebesar 1000 dan faktor pereduksi temperature (cr) sebesar 0,6 dan 0,9. Karena hasil fungsi antara kedua factor pereduksi temperature tersebut sama, maka parameter terbaik akan dilihat dari waktu komputasi yang lebih cepat. Sehingga, pada data uji 1 didapatkan parameter terbaik pada algoritma sebesar 1000 siklus dan pereduksi temperature sebesar 0,6. Setelah didapatkan parameter terbaik, algoritma akan dicoba lagi untuk mengetahui berapa *gap* yang terjadi antara nilai hasil algoritma dan nilai global

optimal menggunakan penyelesaian eksak. Berikut merupakan hasil perhitungan metode eksak.

Tabel 5.10 Hasil Perhitungan Metode Eksak Data Uji 1

Hasil	Nilai
Jumlah <i>Carrier</i> (<i>unit</i>)	1
Total Biaya (\$)	7187.63
Fungsi <i>Multi objective</i>	0.013
Waktu Komputasi (detik)	870
Iterasi	3578609

Berikut merupakan hasil perbandingan model algoritma dengan metode eksak pada data uji 1.

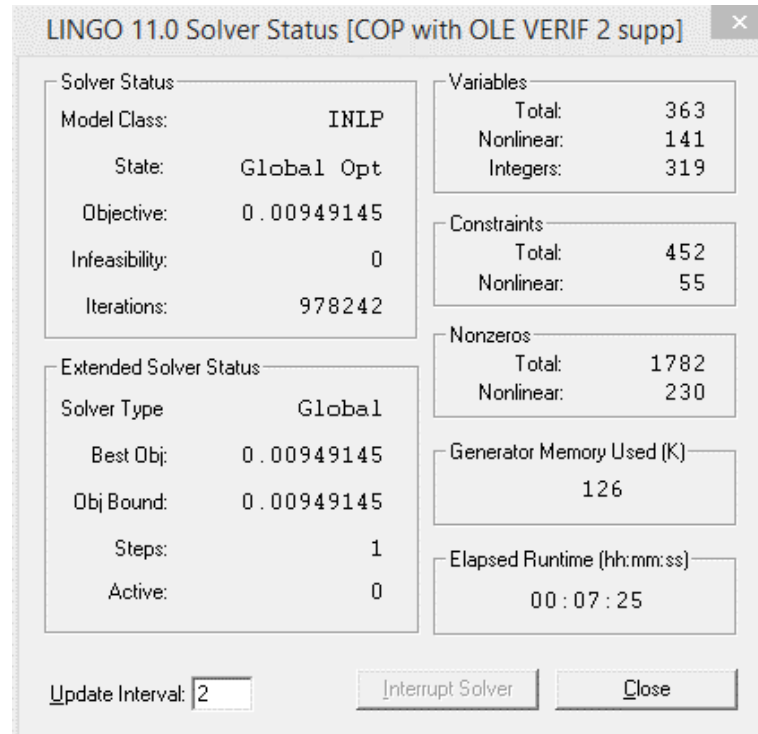
Tabel 5.11 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 1

Replikasi	Jumlah <i>Carrier</i> (<i>unit</i>)		Total Biaya (\$)		<i>Multi objective</i>		Waktu Komputasi (detik)	
	Nilai	Gap%	Nilai	Gap%	Nilai	Gap%	Nilai	Gap%
1	1	0.0%	7187.63	0.0%	0.013	0.0%	40.7	-95.3%
2	1	0.0%	7919.6	10.2%	0.053	292.3%	40.1	-95.4%
3	1	0.0%	7187.63	0.0%	0.013	0.0%	41.7	-95.2%
4	1	0.0%	7187.63	0.0%	0.013	0.0%	40.7	-95.3%
5	1	0.0%	7919.6	10.2%	0.053	292.3%	40.8	-95.3%
6	1	0.0%	7187.63	0.0%	0.013	0.0%	40.9	-95.3%
7	1	0.0%	7187.63	0.0%	0.013	0.0%	41	-95.3%
8	1	0.0%	7187.63	0.0%	0.013	0.0%	41.1	-95.3%
9	1	0.0%	7187.63	0.0%	0.013	0.0%	41.2	-95.3%
10	1	0.0%	7187.63	0.0%	0.013	0.0%	40.9	-95.3%

Dari tabel tersebut didapatkan hasil algoritma tidak sepenuhnya memenuhi global optimum, tetapi dari 10 replikasi, mayoritas dari hasil menunjukkan gap sebesar 0%, sedangkan waktu komputasi menggunakan algoritma SA lebih cepat sekitar 95.3%.

5.3.3 Eksperimen Data Uji 2

Sebelum melakukan uji parameter akan digunakan metode eksak terlebih dahulu untuk mengetahui berapa jumlah variabel dan konstrain yang dibangun. Berikut merupakan hasil fungsi multi tujuan menggunakan metode eksak.



Gambar 5.2 Hasil *Running* Lingo pada Data Uji 2

Pada data dengan ukuran 5 produk dan 3 periode, didapatkan jumlah total variabel yang dicari sebanyak 363 variabel dengan total 452 konstrain. Berikut merupakan hasil yang diperoleh dari hasil *running* model menggunakan data uji 2. Data uji 2 juga digunakan di dalam proses verifikasi dan validasi untuk mengetahui berapa gap yang terjadi.

Tabel 5.12 Hasil Uji Parameter Data Uji 2

Cr	Maxit	Replikasi	Jumlah <i>Carrier</i> (unit)	Total Biaya (\$)	<i>Multi</i> <i>objective</i>	Waktu Komputasi (detik)
0.6	100	1	3	10371	0.282	8.3
		2	2	9644.32	0.009	9
		3	3	8838.4	0.250	8.5
		4	3	8855.8	0.250	8.1
		5	3	8838.4	0.250	8.3
		Rata-rata	2.8	9309.584	0.208	8.44
0.9	100	1	2	9644.32	0.009	36.4
		2	3	8838.4	0.250	36.8
		3	3	8855.8	0.250	35.4
		4	3	8855.8	0.250	38.2
		5	3	8855.8	0.250	36.4

Tabel 5.13 Hasil Uji Parameter Data Uji 2 (Lanjutan)

Cr	Maxit	Replikasi	Jumlah <i>Carrier</i> (unit)	Total Biaya (\$)	<i>Multi</i> <i>objective</i>	Waktu Komputasi (detik)
		Rata-rata	2.8	9010.024	0.202	36.64
0.6	1000	1	2	9644.32	0.009	81
		2	2	9644.32	0.009	81.2
		3	3	8838.4	0.250	82.6
		4	3	8838.4	0.250	82
		5	3	8838.4	0.250	81.6
		Rata-rata	2.6	9160.768	0.154	81.68
0.9	1000	1	2	9644.32	0.009	380.8
		2	3	8838.4	0.250	381.2
		3	3	8855.8	0.250	380.8
		4	2	9644.32	0.009	380.2
		5	3	8855.8	0.250	381.4
		Rata-rata	2.6	9167.728	0.154	380.88

Dari uji parameter data uji 2 didapatkan nilai terbaik dengan parameter factor pereduksi temperature sebesar 0.6 dan maksimum iterasi siklus sebesar 1000. Setelah didapatkan parameter terbaik, algoritma akan dicoba lagi untuk mengetahui berapa *gap* yang terjadi antara nilai hasil algoritma dan nilai global optimal menggunakan penyelesaian eksak. Berikut merupakan hasil perhitungan metode eksak.

Tabel 5.14 Hasil Perhitungan Metode Eksak Data Uji 2

Hasil	Nilai
Jumlah <i>Carrier</i> (unit)	2
Total Biaya (\$)	9644.32
<i>Multi objective</i>	0.009
Waktu Komputasi (detik)	445
Iterasi	978242

Berikut merupakan hasil perbandingan model algoritma dengan metode eksak pada data uji 1.

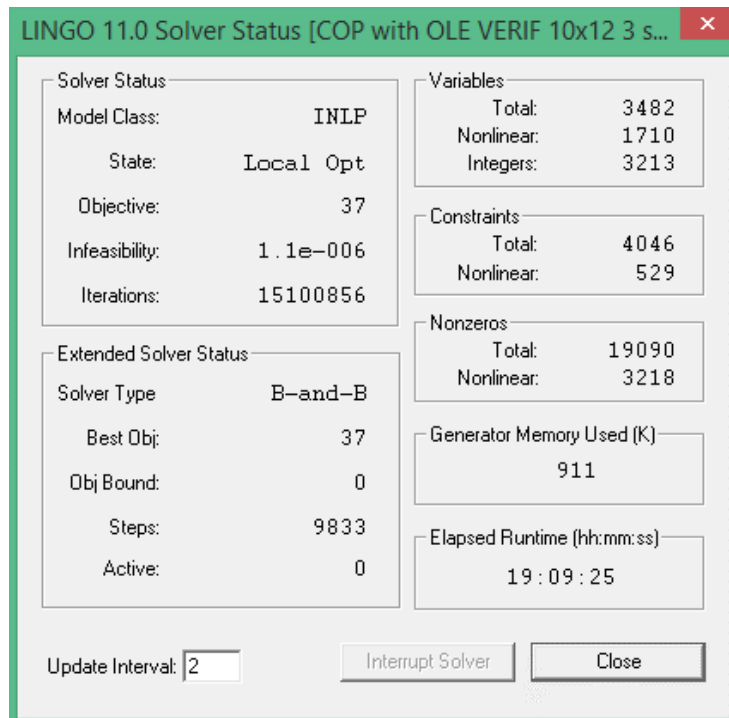
Tabel 5.15 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 2

Replikasi	Jumlah <i>Carrier (unit)</i>		Total Biaya (\$)		<i>Multi objective</i>		Waktu Komputasi (detik)	
	Nilai	Gap%	Nilai	Gap%	Nilai	Gap%	Nilai	Gap%
1	2	0.0%	9644.32	0.0%	0.009	0.0%	40.7	-90.9%
2	2	0.0%	9644.32	0.0%	0.009	0.0%	40.1	-91.0%
3	3	50.0%	8838.4	-8.4%	0.250	2537.4%	41.7	-90.6%
4	3	50.0%	8838.4	-8.4%	0.250	2537.4%	40.7	-90.9%
5	2	0.0%	9644.32	0.0%	0.009	0.0%	40.8	-90.8%
6	2	0.0%	10542	9.3%	0.040	319.9%	40.9	-90.8%
7	2	0.0%	9644.32	0.0%	0.009	0.0%	41	-90.8%
8	2	0.0%	9644.32	0.0%	0.009	0.0%	41.1	-90.8%
9	3	50.0%	8838.4	-8.4%	0.250	2537.4%	41.2	-90.7%
10	2	0.0%	9644.32	0.0%	0.009	0.0%	40.9	-90.8%

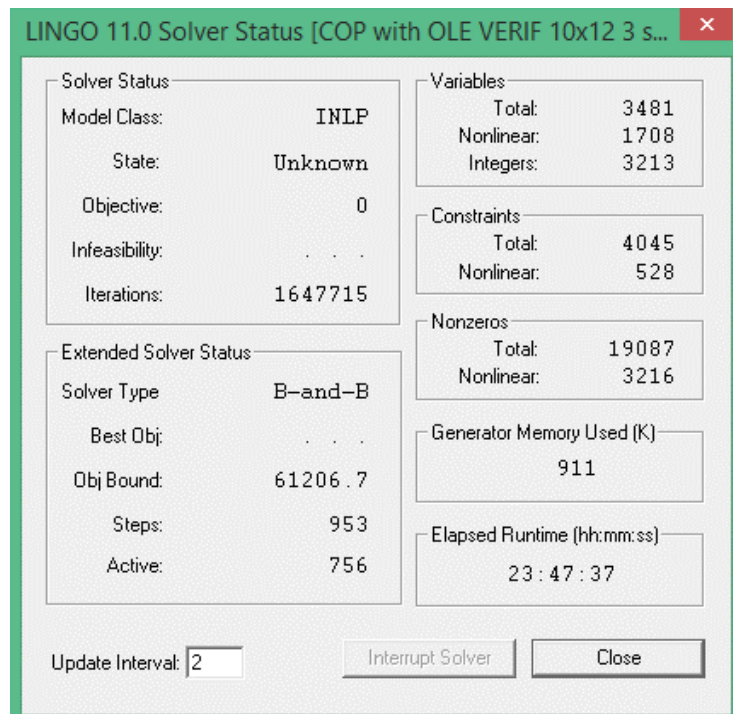
Dari tabel tersebut didapatkan hasil algoritma tidak sepenuhnya memenuhi global optimal, tetapi dari 10 replikasi, mayoritas dari hasil menunjukkan *gap* sebesar 0%, sedangkan waktu komputasi menggunakan algoritma SA lebih cepat sekitar 90.8%.

5.3.4 Eksperimen Data Uji 3

Sebelum dilakukan uji parameter algoritma SA, terlebih dahulu dilakukan running dengan LINGO untuk mengetahui berapa jumlah variabel dan konstrain yang dibangun. Berikut merupakan hasil dari *running* LINGO.



Gambar 5.3 Hasil Running Lingo pada Data Uji 3 Fungsi Tujuan 1



Gambar 5.4 Hasil Running Lingo pada Data Uji 3 Fungsi Tujuan 2

Pada data dengan ukuran 10 produk dan 12 periode, didapatkan jumlah total variabel yang dicari sebanyak 3481 variabel dengan total 4045 konstrain. Pada

fungsi tujuan 1, didapatkan hasil lokal optimal dengan *running* selama 19 jam. Sedangkan pada fungsi tujuan 2, model tidak dapat menemukan nilai fungsi tujuan setelah 23 jam *running*. Oleh karena itu, sebagai perbandingan dengan metode eksak dengan algoritma SA, akan dibandingkan dengan hasil *running* Lingo pada fungsi tujuan 1 dengan hasil lokal optimal.

Berikut merupakan hasil yang diperoleh dari hasil *running* model menggunakan data uji 3. Data uji 3 adalah data uji dengan ukuran 10 *item*, 12 periode, dan 3 *supplier* dari sampel data studi kasus PT INALUM (Persero).

Tabel 5.16 Hasil Uji Parameter Data Uji 3

Cr	Maxit	Replikasi	Jumlah <i>Carrier (unit)</i>	Total Biaya (\$)	Waktu Komputasi (detik)
0.6	100	1	39	116910	866.5
		2	38	119090	880.4
		3	33	125890	896.6
		4	32	124030	938.1
		5	35	122770	989.3
		Rata-rata	35.4	121738	914.2
0.9	100	1	34	116060	1115
		2	33	110390	1394.7
		3	31	115090	1630
		4	32	114680	3562.1
		5	30	105210	1504.2
		Rata-rata	32	112286	1841.2
0.6	1000	1	32	109650	3562.1
		2	32	111260	10521
		3	31	110300	2882.5
		4	29	110240	3809.7
		5	29	107870	4267.9
		Rata-rata	30.6	109864	5008.64
0.9	1000	1	29	107870	5530.4
		2	41	97767	5775.9
		3	33	97764	6044.3
		4	31	112070	19457
		5	37	94814	23457
		Rata-rata	34.2	102057	12052.92

Dari uji parameter data uji 3 didapat nilai fungsi tujuan 1 terbaik pada parameter cr sebesar 0.6 dan maksimum iterasi 1000. Sedangkan pada parameter pereduksi sebesar 0.9 dan maksimum iterasi sebesar 1000, didapatkan rata-rata

fungsi tujuan 2 yang lebih baik dari parameter lainnya. Dari hasil *trade off* tersebut akan diambil parameter dengan waktu komputasi yang lebih rendah yaitu 0.6 dan 1000.

Setelah didapatkan parameter terbaik, algoritma akan dicoba lagi untuk mengetahui berapa *gap* yang terjadi antara nilai hasil algoritma dan nilai optimal menggunakan penyelesaian eksak. Pada hasil metode eksak menggunakan LINGO didapatkan hasil local optimal. Hal ini terjadi karena adanya keterbatasan *memory* dan waktu. Berikut merupakan hasil perhitungan metode eksak.

Tabel 5.17 Hasil Perhitungan Metode Eksak Data Uji 3

Hasil	Nilai
Jumlah <i>Carrier</i> (<i>unit</i>)	37
Total Biaya (\$)	276464.2
Waktu Komputasi (detik)	68965
Iterasi	15100856

Berikut merupakan hasil perbandingan model algoritma dengan metode eksak pada data uji 3.

Tabel 5.18 Hasil Perbandingan Model Algoritma dengan Metode Eksak Data Uji 3

Replikasi	Jumlah <i>Carrier</i> (<i>unit</i>)		Total Biaya (\$)		Waktu Komputasi (detik)	
	Nilai	Gap%	Nilai	Gap%	Nilai	Gap%
1	32	-13.5%	109650	-60.3%	3562.1	-94.8%
2	32	-13.5%	111260	-59.8%	10521	-84.7%
3	31	-16.2%	110300	-60.1%	2882.5	-95.8%
4	29	-21.6%	110240	-60.1%	3809.7	-94.5%
5	29	-21.6%	107870	-61.0%	4267.9	-93.8%
6	33	-10.8%	97764	-64.6%	3962.1	-94.3%
7	31	-16.2%	112070	-59.5%	2052.3	-97.0%
8	37	0.0%	94814	-65.7%	3962.9	-94.3%
9	32	-13.5%	114680	-58.5%	3394.6	-95.1%
10	30	-18.9%	105210	-61.9%	4702.2	-93.2%
Rata-rata	31.6	-14.6%	107385.8	-61.2%	4311.7	-93.7%

Dari tabel tersebut didapatkan hasil algoritma lebih baik daripada menggunakan metode eksak. Tetapi hal tersebut terjadi karena pada hasil metode eksak didapatkan hasil lokal optimal.

5.4 Eksperimen pada Studi Kasus PT INALUM (Persero)

Setelah didapatkan parameter yang menghasilkan solusi terbaik atau lebih cepat konvergen, algoritma SA dicoba untuk menyelesaikan permasalahan riil pada data studi kasus *spare part inventory* PT INALUM (Persero). Berikut merupakan hasil yang didapatkan dari algoritma SA dengan parameter pereduksi temperatur sebesar 0.6 dan maksimum iterasi siklus sebesar 1000.

Tabel 5.19 Hasil *Running Model* pada Penyelesaian Studi Kasus

Replikasi	Jumlah <i>Carrier</i> (unit)	Total Biaya (\$)	Waktu Komputasi (detik)
1	74	512,920	37,031.3
2	76	545,650	37,031.3
3	74	501,610	19,345.7
4	81	705,110	16,634.0
5	77	739,400	21,635.0
6	79	596,350	32,117.0
7	72	645,290	36,708.0
8	74	512,920	37,031.3
9	76	545,650	37,031.3
10	72	645,290	36,708.0
Rata-rata	75.5	595019	31,127.3

Dari hasil tersebut didapatkan hasil optimal dengan jumlah *carrier* sebesar 72 unit, tetapi dengan total biaya sebesar \$645.290. Dan juga hasil optimal dengan biaya sebesar \$501.610 dan jumlah *carrier* sebesar 74 unit. Gambaran contoh skema pemesanan yang dihasilkan dari model yang dibangun pada *supplier* dapat dilihat pada Lampiran F. Berikut merupakan hasil *carrier supplier* yang dihasilkan model optimasi.

Tabel 5.20 Jumlah *Carrier* Hasil Algoritma SA

Supplier	Carrier (unit)
AD2	14
CI10	44
GE16	16
Total	74

Berikut merupakan rincian dari total unit *item* yang dipesan dari hasil algoritma *simulated annealing*.

Tabel 5.21 Jumlah Item yang Dipesan dari Hasil Algoritma

Jumlah Item yang Dipesan		Periode												Total
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
Supplier	CI10	395	29	23	142	46	31	76	17	42	19	19	0	839
	AD2	579	25	37	41	137	36	72	2	26	0	7	0	962
	GE16	870	762	21	144	36	15	61	27	7	9	4	5	1961
Total		1844	816	81	327	219	82	209	46	75	28	30	5	3762

Jika dilihat pada tabel di atas, mayoritas pesanan dilakukan di awal-awal periode. Sedangkan jumlah pesanan pada akhir periode jumlah pesannya semakin sedikit. Setelah didapatkan jumlah *carrier* dan jumlah *item* yang dipesan, juga dilakukan perhitungan pada fungsi tujuan 2 yaitu biaya persediaan. Berikut merupakan rincian komponen biaya yang dihasilkan model.

Tabel 5.22 Hasil Biaya Algoritma SA

Biaya	Nilai
Biaya Pemesanan	\$ 4,994.80
Biaya Pembelian	\$ 358,750.00
Biaya Penyimpanan	\$ 140,870.00
Total Biaya	\$ 504,614.80

Dari hasil tersebut dapat diketahui dari 74 *carrier*, 44 diantaranya merupakan *carrier* dari *supplier* CI10, serta biaya terendah terletak pada biaya pemesanan. Sedangkan pada biaya *backorder* nilainya sebesar \$0, karena tidak terjadi *shortage*. Hasil ini kemudian akan dibandingkan dengan kondisi eksisting yang diterapkan perusahaan.

5.5 Analisis Hasil Eksperimen

Pada subbab analisis dibahas mengenai analisis kondisi eksisting, analisis eksperimen uji parameter, analisis hasil, dan perbandingan hasil dengan kondisi eksisting perusahaan.

5.5.1 Analisis Kondisi Eksisting

Pada kondisi eksisting perusahaan menerapkan kebijakan *re-order level* sebesar *safety stock* dengan jumlah pemesanan sebesar kumulatif permintaan selama 3 bulan kedepan. Selain itu, pemesanan tiap *item* juga dilakukan secara terpisah tidak melihat historis pembelian *supplier* perusahaan. Pemesanan *spare part* yang dilakukan secara independen pada PT INALUM (Persero) menyebabkan tingginya *ordering cost*. Padahal, proses tender pada *spare part* tidak terlalu signifikan berpengaruh, karena harus melakukan pembelian *item* yang banyak jenisnya dan penawaran dilihat berdasarkan kesesuaian spesifikasi *part*. Jadi, saat ada *item* dengan *supplier* yang sama, sementara kebutuhannya pada periode yang berbeda, *item – item* tersebut akan dipesan secara independen pada periode dimana *item* tersebut dibutuhkan. Hal ini menyebabkan jumlah *carrier* yang diantarkan *supplier* menjadi tinggi karena bila pemesanan yang dilakukan berbeda periodenya. Padahal, volume angkut masih jauh dari kapasitas dan menyebabkan utilitas *carrier* menjadi rendah dan waktu pengiriman yang lebih lama.

Selain itu, adanya biaya *inventory* lainnya seperti *holding cost* harus dipertimbangkan. Karena jika dilakukan terlalu banyak pesanan di waktu yang sama, *holding cost* yang dikenai menjadi tinggi. Dari hasil perhitungan, total biaya yang dikeluarkan perusahaan sebesar \$586,185.59 dan total jumlah *carrier supplier* sebesar 76 unit.

5.5.2 Analisis Eksperimen Uji Parameter Algoritma SA

Eksperimen yang dilakukan pada algoritma SA, didapatkan parameter terbaik sebesar 1000 maksimum siklus dan 0.6 pereduksi temperatur. Jika dilihat pada hasil tiap parameter, semakin lama waktu komputasi atau semakin besar jumlah iterasi, solusi yang didapatkan bisa jauh lebih optimal.

Jika digunakan maksimum 1000 iterasi tiap siklus, hasilnya jauh lebih baik dari pada menggunakan 100 iterasi. Sama halnya pada faktor pereduksi temperatur, semakin besar nilai *cr* hasil yang didapatkan bias jauh lebih baik. Hal ini dikarenakan dengan jumlah iterasi yang kecil, solusi yang dihasilkan algoritma SA memiliki peluang yang lebih besar untuk terjebak di lokal optimal. Walaupun semua parameter hasilnya lebih baik dari pada solusi awal yang dibangkitkan.

Perbedaan solusi awal yang dibangkitkan secara acak juga mempengaruhi hasil algoritma sehingga memungkinkan solusi satu dengan lainnya tidak konvergen.

Pada data uji 1 dan 2, solusi yang dihasilkan lebih mudah mencapai nilai optimalnya karena ukuran data yang kecil. Tetapi saat dicoba dengan ukuran data yang lebih besar pada data uji 3, hasil setiap replikasi bisa jauh berbeda. Bahkan, saat dilakukan eksperimen menggunakan metode eksak, hasilnya tidak bisa mencapai nilai optimalnya karena keterbatasan *memory* dan waktu. Hal ini terjadi karena model permasalahan terlalu kompleks. Untuk ukuran 10 item saja, jumlah total variabel yang dicari sebanyak 3481 variabel dengan total 4045 konstrain. Sehingga untuk menemukan solusi yang optimal sangat susah. Jika dilihat hasil perhitungan eksak, setelah *running* model dengan fungsi tujuan 2 selama lebih dari 23 jam, model tersebut masih tidak bisa menemukan solusi optimal yang *feasible*. Jadi, untuk mendapatkan solusi yang optimal menggunakan algoritma SA pun membutuhkan waktu yang sangat lama mencapai 10 jam. Itupun didapatkan hasil dengan *gap* yang cukup tinggi jika dilihat kembali pada hasil data uji 1 dan 2. Jika bisa dilakukan lebih banyak iterasi lagi, bisa jadi model yang dibangun dapat mendapatkan solusi terbaiknya.

5.5.3 Analisis Hasil Penyelesaian Studi Kasus

Pada permasalahan dengan jumlah data yang besar yaitu 157 *item*, 12 periode, dan 3 *supplier*, hasil setiap replikasinya bisa jauh berbeda. Hal ini terjadi karena model diperlukan jumlah pembangkitan solusi yang sangat besar, yaitu 49.455 variabel keputusan di setiap iterasinya. Banyaknya variabel tersebut terjadi karena banyaknya jumlah item yang berkorelasi satu dengan lainnya. Sehingga setiap iterasi diperlukan waktu dari 5 hingga 10 detik. Saat dicoba dengan parameter yang optimal, waktu penyelesaiannya pun menjadi sangat lama, yaitu lebih dari 10 jam. Tetapi, jika dibandingkan dengan penyelesaian eksak, pada data uji dengan jumlah item sebanyak 10 saja, waktu penyelesaiannya bisa lebih dari 19 jam dan hanya mencapai lokal optimal.

Untuk menyelesaikan data dengan jumlah besar diperlukan *hardware* dengan *memory* dan *processor* yang besar agar didapatkan solusi yang lebih cepat. Sehingga total iterasinya bisa ditambahkan agar didapatkan solusi yang lebih baik.

Karena, apabila jumlah iterasi pada algoritma SA diperbanyak, bisa jadi didapatkan solusi yang jauh lebih baik dari pada solusi yang telah didapatkan.

Dari hasil model algoritma didapatkan minimum biaya sebesar \$501,610 dan total *carrier* sebesar 74 unit. Dari hasil tersebut dapat diketahui dari 74 *carrier*, 44 diantaranya merupakan *carrier* dari *supplier* CI10, serta biaya terendah terletak pada biaya pemesanan. Hasil ini kemudian akan dibandingkan dengan kondisi eksisting yang diterapkan perusahaan. Hasil ini masih lebih baik jika dibandingkan dengan kondisi eksisting yang akan dibahas di Subbab 5.5.4.

5.5.4 Perbandingan Hasil Algoritma SA dengan Kondisi Eksisting

Setelah dilakukan perhitungan kondisi eksisting dan model optimasi, didapatkan hasil perbaikan seperti pada tabel berikut ini.

Tabel 5.23 Perbandingan Hasil Kondisi Eksisting dan Hasil Optimasi

Fungsi Tujuan	Kondisi Eksisting	Hasil Optimasi	<i>Saving %</i>
Total Biaya	\$ 586,185.59	\$ 501,610.00	14.43%
Total <i>Carrier</i>	76	74	2.63%

Biaya yang dihasilkan menurun 11.09% dari kondisi eksisting dan jumlah *carrier* juga menurun dari 76 unit menjadi 74 unit. Hal ini terjadi karena pada model *can-order policy* dilakukan penyatuan pesanan yang menyebabkan biaya yang lebih rendah. Jika dilihat lebih rinci, berikut merupakan hasil perubahan *carrier supplier*.

Tabel 5.24 Perbandingan Jumlah *Carrier* Kondisi Eksisting dan Hasil Optimasi

<i>Supplier</i>	Carrier (unit)		<i>Saving %</i>
	Kondisi Eksisting	Hasil Optimasi	
AD2	13	14	-0.08%
CI10	41	44	-0.07%
GE16	22	16	27.27%
Total	76	74	2.63%

Hasil optimasi menunjukkan bahwa pada *supplier* AD2 dan CI10, hasilnya lebih buruk dari pada kondisi eksisting karena jumlah *carrier*-nya yang semakin meningkat sebanyak 1 unit pada *supplier* AD2 dan 3 unit pada *supplier* CI10. Tetapi, pada *supplier* GE16 memiliki penurunan jumlah yang signifikan sebanyak

27.27% dari kondisi awal. Hasil ini juga menunjukkan adanya perbedaan *treatment* di setiap *supplier*. Bisa jadi karena volume dan jumlah pemesanan *supplier* AD2 dan CI10 tinggi, sehingga jumlah *carrier* juga meningkat. Secara total, jumlah *carrier* menurun dari 76 unit menjadi 74 unit.

Jika dilihat pada jumlah pemesanan *item* dari kondisi eksisting pada Tabel 5.7 dan jumlah unit pemesanan pada Tabel 5.22, adanya jumlah pemesanan *item* yang berbeda. Dari total 4132 *item* pada kondisi eksisting, berkurang 8,9% (370 unit) menjadi 3762 unit *item*. Pola pemesanan kondisi eksisting dan hasil optimasi cenderung sama, yaitu lebih banyak melakukan pemesanan di awal-awal periode. Perbedaan ini terjadi karena adanya perbedaan kebijakan mengenai *re-order level* dan jumlah pemesanan atau *order-up-to level*. Pada kondisi eksisting, terdapat beberapa *item* yang *re-order level*-nya lebih besar jika dibandingkan dengan hasil perbaikan. Sehingga, pada periode-periode akhir, seperti pada bulan Juli dan Oktober, jumlah pemesanan kondisi eksisting bisa jauh lebih banyak dari pada hasil perbaikan. Kemudian juga dibandingkan setiap komponen biaya yang dikeluarkan perusahaan sebagai berikut.

Tabel 5.25 Perbandingan Biaya Kondisi Eksisting dan Hasil Optimasi

Komponen Biaya	Biaya (\$)		Saving %
	Kondisi Eksisting	Hasil Optimasi	
Biaya Pemesanan	\$ 20,756.84	\$ 4,994.80	75.94%
Biaya Pembelian	\$ 406,771.11	\$ 358,750.00	11.81%
Biaya Penyimpanan	\$ 158,657.64	\$ 140,870.00	11.21%
Total Biaya	\$ 586,185.59	\$ 504,614.80	14.43%

Jika dilihat dari komponen setiap biaya, perubahan yang signifikan terjadi pada biaya pemesanan yaitu menurun sebesar 75.94%, hal ini terjadi karena pada model *multiple can-order level* yang dibangun memungkinkan terjadinya penyatuan pesanan pada setiap kelompok *item* yang memiliki kesamaan *supplier*. Sehingga biaya pemesanan major yang dikenakan pada *item* yang dipesan dapat turun secara signifikan. Sedangkan pada total harga *item* dan biaya penyimpanan menurun sebesar 11.81% dan 11.21%. Penurunan biaya pembelian terjadi karena adanya perbedaan jumlah pesanan antara kondisi eksisting dan hasil perbaikan. Perbedaan kebijakan *re-order level* dan *order quantity* menyebabkan jumlah *item* yang dipesan juga menjadi berbeda. Hal tersebut juga mempengaruhi biaya

penyimpanan yang juga menurun. Pada biaya *backorder*, nilai kondisi eksisting dan optimasi sebesar \$0. Jika dilihat pada total biaya, penghematan yang terjadi sebesar 14.43% dari kondisi eksisting atau sebesar \$81,570.79 (Rp 1,060,420,270). Dengan hasil seperti ini, dapat dikatakan bahwa model dan algoritma SA yang telah dibangun dapat mencapai solusi yang lebih baik dari kondisi awal.

(Halaman Ini Sengaja Dikosongkan)

BAB 6

KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan hasil penelitian dan saran-saran yang berkaitan dengan penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, berikut ini merupakan kesimpulan yang dapat diambil :

1. Berdasarkan eksperimen dengan 3 *supplier* dan 157 *item* menggunakan *multiple can-order level* dengan *can-order policy* yang dihasilkan algoritma *Simulated Annealing*, didapatkan hasil yang lebih baik dari kebijakan persediaan di PT INALUM (Persero) dengan biaya persediaan sebesar \$501,610.00 dan jumlah *carrier* sebanyak 74 unit. Hasil tersebut menunjukkan adanya penghematan dari kondisi eksisting yang diterapkan perusahaan sebesar 2 unit pada jumlah *carrier*, dan penghematan biaya 14.43% atau sebesar \$81,570.79 (Rp 1,060,420,270).
2. Berdasarkan hasil eksperimen yang telah dilakukan, dapat diketahui bahwa algoritma SA yang dibangun dapat menemukan nilai solusi yang mendekati optimum, tetapi sulit mencapai nilai global optimal bila dilakukan *running* model dengan jumlah data yang besar. Pada eksperimen yang dilakukan, *gap* yang terjadi antara hasil algoritma SA dengan perhitungan eksak masih berselisih jauh. Hal ini terjadi karena saat dilakukan dalam jumlah yang besar, variabel yang dibangkitkan nilainya juga semakin besar dan meningkat secara eksponensial. Sehingga waktu komputasi yang dihasilkan juga menjadi sangat lama. Untuk mengatasi hal tersebut dapat, dapat dilakukan *running* model dengan jumlah iterasi yang besar, yaitu dengan menaikkan nilai reduksi temperatur maupun menaikkan nilai maksimum iterasi siklus.

6.2 Saran

Berikut ini adalah beberapa saran yang dapat diberikan untuk penelitian selanjutnya :

1. Melakukan *running* model algoritma SA dengan jumlah iterasi yang besar dan memperbaiki cara algoritma menemukan solusinya agar didapatkan hasil dan waktu yang lebih baik atau dengan mencoba algoritma metaheuristic dan heuristic lainnya.
2. Penyelesaian permasalahan *multi objective* dapat diselesaikan dengan pendekatan metode lainnya, karena pada metode global kriteria memiliki waktu komputasi yang lama karena akan dibuat fungsi tujuan baru berbentuk *non-linear*.
3. Dalam pengembangan model selanjutnya, dapat dipertimbangkan faktor-faktor lain seperti *discount rate*, perubahan harga akibat inflasi, biaya transportasi, dan lain-lain atau dengan nilai variabel yang stokastik, misalnya pada *demand* dan *lead time*.

DAFTAR PUSTAKA

- Arnold, T., Chapman, S. & Clive, L., 2008. *Introduction to Materials Management*. 6th ed. New Jersey: Pearson Education, Inc.,.
- Atkins, D. R. & Iyogun, P. O., 1988. Periodic Versus “Can-Order” Policies for Coordinated Multi-Item Inventory Systems. *Management Science*, 34(6).
- Balintfy, J. L., 1964. On A Basic Class of Multi-Item Inventory Problems. *Manage. Sci.*, Volume 10, pp. 287-297.
- Ballou, R. H., 2004. *Business Logistics / Supply Chain Management*. 5th ed. New Jersey: Pearson/Prentice Hall Inc..
- Cohon, J. J. & Marks, D. H., 1975. A review and evaluation of multiobjective programming techniques. *Water Resour. Res* 11, pp. 208-220.
- Eaves, A. C. & Kingsman, B. G., 2004. Forecasting for the ordering and stockholding of spare parts. *Journal of the Operational Research Society*, Issue 55, pp. 431-437.
- Federgruen, A., Groenevelt, H. & Tijms, H. C., 1984. Coordinated Replenishment in a Multi-Item Inventory System with Compound Poisson Demands. *Manage. Sci.*, Volume 30, pp. 344-357.
- Fogarty, D. W., Blackstone, J. H. & Hoffmann, T. R., 1991. *Production & Inventory Management*. 2nd ed. Cincinnati, Ohio: South-Western Publishing Co..
- Fransiska, L., 2011. *Pengendalian Persediaan Komponen Circuit Breaker Dengan Kebijakan Can-Order (Studi Kasus : PT.E-T-A Indonesia)*, Surabaya: Departemen Teknik Industri FTI-ITS.
- Ghorbel, N., Addouche, S.-A. & El-Mhamedi, A., 2014. Replenishment Policies in Static and Dynamic Spare Part Inventory Control: A Survey. *IJEDR ISSN: 2321-9939*, 2(4).
- Heinecke, G., Syntetos, A. A. & Wang, W., 2013. Forecasting-based SKU classification. *International Journal of Production Economics*, Issue 143 (2), pp. 455-462.
- Jamalnia, A. & Soukhakian, M. A., 2008. A hybrid fuzzy goal programming approach with different goal priorities. *Elsevier Computer & Industrial Engineering*, Volume 56.

- Johansen, S. G. & Melchior, P., 2003. Can-Order Policy for the Periodic-Review Joint Replenishment Problem. *Journals of the Operational Research Society*, Issue 54, pp. 283-290.
- Nagasawa, K. & Irohara, T., 2016. Multiple Can-order Level for Can-order Policies under Carrier Capacity and Correlated Demands. *J Jpn Ind Manage Assoc* 67, pp. 114-123.
- Porras, E. & Dekker, R., 2008. A solution method for the joint replenishment problem with correction factor. *International Journal of Production Economics*, Issue 113(2), pp. 834-851.
- Putra, I. A. & Pujawan, I. N., 2011. *Pengendalian Persediaan Spare Part dengan Menggunakan Can-Ordering Policy Studi Kasus: PT. PJB Unit Pembangunan Gresik*, Surabaya: Jurusan Teknik Industri Institut Teknologi Sepuluh Nopember.
- Ramaekers, K. & Janssens, G. K., 2014. Optimal Policies for Demand Forecasting and Inventory Management of Goods with Intermittent Demand. *Journal of Applied Operational Research* 6(2), pp. 111-123.
- Santosa, B. & Willy, P., 2011. *Metoda Metaheuristik Konsep dan Implementasi*. 1st Edition ed. Surabaya: Prima Printing.
- Sipper, D. & Bulfin, R., 1997. *Production: Planning, Control, and Integration*. United States of America: The McGraw-Hill Companies, Inc.
- Syntetos AA, B. J. d. C. J., 2005. On the Categorization of Demand Patterns. *J Opl Res Soc* 56: 495–503.
- Tabucanon, M. T., 1988. *Multiple Criteria Decision Making in Industry*. Amsterdam: Elsevier.
- Vollmann, T. E., Berry, W. L. & Whybark, D. C., 1997. *Manufacturing Planning and Control System 4th Edition*. United States of America: The McGraw-Hill Companies, Inc.
- Vrat, P., 2014. *Materials Management, Springer Texts in Business and Economics*. DOI 10.1007/978-81-322-1970-5_2 ed. India: Springer India.
- Williams, T. M., 1984. Stock control with sporadic and slow-moving demand. *Journal of the Operational Research Society*, 35(10), pp. 939-948.

LAMPIRAN A

DATA UJI

Berikut merupakan data uji yang digunakan dalam proses uji parameter algoritma SA.

Data Uji 2 (5 *item*, 3 periode, 2 *supplier*)

Material Number	Demand			Supplier Code	Initial Inventory	Safety Stock	Price	Satuan	Critical Code	Volume	Holding Cost	Penalty	Backorder Cost
	t1	t2	t3										
G1295	0	4	4	GE16	4	8	\$ 47.54	each	B	105	\$ 4.99	30%	\$ 61.80
G1043	21	3	3	GE16	1	13	\$ 40.80	each	A	5.04	\$ 4.28	50%	\$ 61.20
G0100	3	7	0	GE16	1	6	\$ 146.17	each	A	157.383	\$ 15.35	50%	\$ 219.26
C0823	1	0	3	CI10	1	2	\$ 272.13	each	C	1841.508	\$ 28.57	15%	\$ 312.95
C0075	8	4	4	CI10	6	16	\$ 516.24	each	A	116.28	\$ 54.21	50%	\$ 774.36

Data Uji 3 (10 *item*, 12 periode, 3 *supplier*)

Material Number	Demand												
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	2018 Total
A2276	0	0	6	2	4	0	0	2	0	0	0	0	14
A4311	76	14	14	14	14	14	16	0	0	0	0	0	162
A0161	0	2	2	1	10	0	0	0	0	0	0	0	15
C0823	1	0	3	3	0	0	0	0	0	1	0	0	8
C1243	0	0	0	0	1	1	2	0	0	0	1	0	5
C0075	8	4	4	12	0	4	8	0	0	0	4	0	44
G0288	0	0	0	3	7	0	3	5	3	2	2	0	25
G1043	14	7	7	21	3	3	6	0	0	0	3	0	64
G0100	0	0	0	3	7	0	3	5	3	2	2	0	25
G0089	7	0	7	14	7	0	0	0	0	0	0	0	35

Material Number	Kode Vendor	Critical Code	Volume	Initial Inventory	Holding Cost	Minor Order Cost	Penalty	Backorder Cost	Safety Stock	Harga	Satuan
A2276	AD2	B	141354.18	2	\$ 13.43	\$ 2.69	30%	\$ 174.58	5	\$ 134.29	each
A4311	AD2	A	958.3	2	\$ 6.99	\$ 1.40	50%	\$ 104.90	13	\$ 69.93	each
A0161	AD2	C	360	6	\$ 1.98	\$ 0.40	15%	\$ 22.75	6	\$ 19.78	set
C0823	CI10	C	408.32	1	\$ 27.21	\$ 5.44	15%	\$ 312.95	2	\$ 272.13	each
C1243	CI10	A	2700000	0	\$ 271.68	\$ 54.34	50%	\$ 4,075.25	3	\$ 2,716.83	each
C0075	CI10	A	1162.8	6	\$ 51.62	\$ 10.32	50%	\$ 774.36	16	\$ 516.24	each
G0288	GE16	B	5877.23	1	\$ 7.29	\$ 1.46	30%	\$ 94.80	5	\$ 72.92	each
G1043	GE16	A	1176	1	\$ 4.08	\$ 0.82	50%	\$ 61.20	13	\$ 40.80	each
G0100	GE16	A	544.5	1	\$ 14.62	\$ 2.92	50%	\$ 219.26	6	\$ 146.17	each
G0089	GE16	C	56000	7	\$ 1.30	\$ 0.26	15%	\$ 14.97	5	\$ 13.02	each

LAMPIRAN B

DATA STUDI KASUS *SPARE PART INVENTORY* PT INALUM

(Persero)

Berikut merupakan data yang digunakan untuk menyelesaikan permasalahan perusahaan sebanyak 157 item, 12 periode, dan 3 *supplier*.

No	Material Number	Demand												2018 Total
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
1	A0116	0	14	0	0	0	0	0	0	0	0	0	0	14
2	A0391	0	0	0	0	0	0	24	0	0	0	0	0	24
3	A0408	8	0	0	1	0	0	2	0	0	0	0	0	11
4	A0417	0	0	0	49	1	1	1	0	0	16	0	0	68
5	A0419	0	0	0	0	0	0	0	0	1	0	0	0	1
6	A0510	0	0	0	1	0	0	0	0	0	0	0	0	1
7	A0531	0	0	0	0	0	2	0	0	0	0	0	0	2
8	A0533	0	0	0	0	0	0	24	0	0	0	0	0	24
9	A0650	3	1	0	3	2	0	0	0	0	0	0	0	9
10	A0668	0	0	0	0	0	4	0	0	0	0	0	0	4
11	A0812	0	0	0	1	0	0	1	0	0	0	0	0	2
12	A1025	0	2	0	0	2	0	2	2	0	0	0	0	8
13	A1216	8	0	0	4	0	0	0	0	0	0	0	0	12
14	A0022	0	0	0	2	0	6	2	0	0	0	0	0	10
15	A1857	0	0	0	4	0	0	0	0	0	0	0	0	4
16	A1860	4	0	1	0	0	0	0	0	0	0	0	0	5
17	A1876	0	0	0	2	0	0	0	0	0	0	0	0	2
18	A1910	1	0	8	12	3	5	0	1	1	0	0	0	31
19	A2276	0	0	6	2	4	0	0	2	0	0	0	0	14
20	A2300	0	0	0	0	0	1	0	0	1	0	0	0	2
21	A2319	0	0	0	0	0	2	0	0	2	0	0	0	4
22	A2322	0	0	0	0	1	0	0	1	0	0	0	0	2
23	A2330	0	0	0	0	2	0	0	2	0	0	0	0	4
24	A2753	0	0	0	0	0	4	0	0	0	0	0	0	4
25	A0185	15	0	0	0	0	0	0	0	0	0	0	0	15
26	A0704	1	0	1	0	0	0	0	0	0	0	0	0	2
27	A0705	3	0	0	0	0	0	0	0	0	0	0	0	3
28	A0742	1	0	1	0	0	0	0	0	0	0	0	0	2
29	A0753	2	0	2	0	0	0	0	0	0	0	0	0	4
30	A0099	1	0	0	1	0	0	0	0	0	0	0	0	2
31	A2230	0	0	0	1	0	8	16	0	0	0	8	0	33
32	A2600	1	0	0	8	0	0	0	0	0	0	0	0	9

No	Material Number	Demand												
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	2018 Total
33	A3336	0	0	0	0	1	0	0	1	0	0	0	0	2
34	A4311	76	14	14	14	14	14	16	0	0	0	0	0	162
35	A0135	0	2	0	0	0	0	0	0	0	0	0	0	2
36	A0144	0	0	11	0	60	0	0	0	0	0	0	0	71
37	A0147	0	2	0	0	0	0	0	0	0	0	0	0	2
38	A0150	0	4	0	0	0	0	0	0	0	0	0	0	4
39	A0074	0	0	0	0	0	1	0	0	0	1	0	0	2
40	A0120	0	0	0	0	0	2	4	0	0	0	2	0	8
41	A0134	0	1	0	0	0	0	0	0	0	0	0	0	1
42	A0158	1	0	0	2	0	0	0	0	0	0	0	0	3
43	A0161	0	2	2	1	10	0	0	0	0	0	0	0	15
44	A0162	0	1	0	1	0	0	0	0	0	0	0	0	2
45	A0171	0	0	0	1	0	0	0	0	0	0	0	0	1
46	A0176	0	4	0	0	0	0	0	0	0	0	0	0	4
47	A0060	0	0	4	0	0	0	0	0	0	0	0	0	4
48	A0154	8	4	4	12	0	0	0	0	0	0	0	0	28
49	A0237	0	0	4	0	0	0	0	0	0	0	0	0	4
50	A0090	0	0	0	0	1	1	2	0	0	0	1	0	5
51	A0618	16	16	0	0	0	0	16	0	0	0	0	0	48
52	A0934	0	0	0	0	0	0	0	0	38	0	0	0	38
53	C0818	0	0	0	0	2	2	3	0	0	0	0	0	7
54	C0822	0	1	0	2	3	1	0	0	0	0	0	0	7
55	C0823	1	0	3	3	0	0	0	0	0	1	0	0	8
56	C0848	0	0	0	0	2	0	2	0	0	0	0	0	4
57	C1314	0	0	0	0	0	0	4	0	0	0	0	0	4
58	C1362	0	0	0	1	1	0	0	0	0	0	0	0	2
59	C1364	0	0	0	6	6	4	0	0	0	0	0	0	16
60	C1366	0	0	0	2	2	0	0	0	0	0	0	0	4
61	C1369	0	0	0	0	0	8	0	0	0	0	0	0	8
62	C1382	0	0	0	2	0	2	0	0	1	1	0	0	6
63	C0206	0	9	20	9	0	0	0	0	0	0	0	0	38
64	C0995	0	0	1	0	0	0	0	0	0	0	0	0	1
65	C1486	0	0	1	0	0	0	0	0	0	0	0	0	1
66	C1488	0	0	1	0	0	0	0	0	0	0	0	0	1
67	C2211	0	0	0	0	0	4	0	0	0	0	0	0	4
68	C2226	8	6	2	0	0	0	0	0	0	0	0	0	16
69	C2422	0	0	1	0	0	0	0	0	0	0	0	0	1
70	C2425	1	0	0	0	1	0	0	0	0	0	0	0	2
71	C2503	0	1	1	0	0	0	0	0	0	0	0	0	2
72	C0255	0	0	20	0	0	0	0	0	0	0	0	0	20

No	Material Number	Demand												2018 Total
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
73	C0259	0	0	0	0	76	0	0	0	0	0	0	0	76
74	C0061	0	0	3	0	0	0	0	0	0	0	0	0	3
75	C0002	0	0	0	0	1	1	2	0	0	0	1	0	5
76	C1243	0	0	0	0	1	1	2	0	0	0	1	0	5
77	C1945	0	0	0	0	0	0	2	0	0	0	0	0	2
78	C2759	0	0	0	1	0	0	0	0	0	0	0	0	1
79	C2845	0	0	0	0	0	0	0	0	24	0	0	0	24
80	C2871	4	0	0	0	8	0	0	0	0	0	0	0	12
81	C3079	0	6	0	0	0	0	0	0	0	0	0	0	6
82	C3289	0	0	0	0	2	0	0	0	0	0	0	0	2
83	C3462	0	0	0	0	40	0	40	0	0	0	0	0	80
84	C3613	0	1	0	0	0	0	0	0	0	0	0	0	1
85	C3640	15	0	0	0	30	0	0	0	0	0	0	0	45
86	C4119	2	0	0	8	0	0	0	0	0	0	0	0	10
87	C0072	0	0	1	0	0	0	0	0	0	0	0	0	1
88	C0273	3	0	6	0	6	0	0	0	0	0	0	0	15
89	C0018	2	0	0	0	1	0	0	0	0	0	0	0	3
90	C0075	8	4	4	12	0	4	8	0	0	0	4	0	44
91	C0269	9	0	0	0	0	0	0	0	0	0	0	0	9
92	C0281	340	0	0	0	0	0	0	0	0	0	0	0	340
93	C0035	0	0	1	0	0	0	0	0	0	0	0	0	1
94	C0074	0	0	1	0	0	0	0	0	0	0	0	0	1
95	C0225	0	0	0	0	0	0	1	0	0	0	0	0	1
96	G1203	0	0	0	0	0	0	1	0	0	0	0	0	1
97	G1218	0	0	0	0	0	0	0	4	0	0	0	0	4
98	G1295	0	0	0	0	4	4	8	0	0	0	4	0	20
99	G1297	0	0	0	0	4	4	8	0	0	0	4	0	20
100	G0286	0	0	0	1	0	0	0	0	0	0	0	0	1
101	G0472	0	0	0	0	0	0	1	0	0	0	0	0	1
102	G0473	0	0	0	0	0	0	2	0	0	0	0	0	2
103	G0475	0	0	0	2	0	0	0	0	0	0	0	0	2
104	G0476	0	0	0	1	0	0	0	0	0	0	0	0	1
105	G0595	0	0	0	1	0	0	0	0	0	0	0	0	1
106	G0596	0	0	0	1	0	0	0	0	0	0	0	0	1
107	G0597	0	0	0	1	0	0	0	0	0	0	0	0	1
108	G0598	0	0	0	1	0	0	0	0	0	0	0	0	1
109	G0599	0	0	0	2	0	0	0	0	0	0	0	0	2
110	G0732	0	0	0	1	0	0	0	0	0	0	0	0	1
111	G0733	0	0	0	1	0	0	0	0	0	0	0	0	1
112	G0734	0	0	0	4	0	0	0	0	0	0	0	0	4

No	Material Number	Demand												2018 Total
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
113	G0835	0	0	2	0	0	0	0	0	0	0	0	0	2
114	G1095	0	0	0	0	0	0	2	0	0	0	0	0	2
115	G1878	0	0	0	0	1	0	0	0	0	0	0	0	1
116	G2002	0	0	0	12	0	0	0	0	0	0	0	0	12
117	G2004	0	0	0	12	0	0	0	0	0	0	0	0	12
118	G2343	0	0	0	1	0	0	0	0	0	0	0	0	1
119	G2415	0	0	0	0	0	0	1	0	0	0	0	0	1
120	G0288	0	0	0	3	7	0	3	5	3	2	2	0	25
121	G0023	0	0	0	0	0	0	2	0	0	0	0	0	2
122	G0062	0	0	0	1	1	0	0	0	0	0	0	0	2
123	G0109	0	0	0	0	1	1	2	0	0	0	1	0	5
124	G0110	0	0	0	0	0	1	2	0	0	0	1	0	4
125	G0181	0	0	0	3	0	0	0	0	0	0	0	0	3
126	G0212	0	0	0	0	0	0	1	0	0	0	0	0	1
127	G0213	0	0	0	0	0	0	1	0	0	0	0	0	1
128	G0164	0	75 2	0	0	0	0	0	0	0	0	0	0	752
129	G0171	109 6	0	0	0	0	0	0	0	0	0	0	0	1096
130	G0765	0	0	0	0	0	0	2	0	0	0	0	0	2
131	G1043	14	7	7	21	3	3	6	0	0	0	3	0	64
132	G2197	72	0	72	0	0	0	0	0	0	0	0	0	144
133	G2427	0	0	0	0	0	0	1	0	0	0	0	0	1
134	G2428	0	0	0	0	0	0	1	0	0	0	0	0	1
135	G2517	0	0	0	2	0	0	0	0	0	0	0	0	2
136	G2978	0	0	0	0	0	0	1	0	0	0	0	0	1
137	G3063	0	0	0	8	0	0	0	0	0	0	0	0	8
138	G3131	0	0	0	12	0	0	0	0	0	0	0	0	12
139	G3135	0	0	0	0	0	0	1	0	0	0	0	0	1
140	G3142	0	0	0	0	0	0	0	0	4	0	0	0	4
141	G3775	0	0	0	0	0	0	2	0	0	0	0	0	2
142	G4248	0	0	0	0	0	0	1	0	0	0	0	0	1
143	G4391	0	0	0	18	0	0	0	0	0	0	0	0	18
144	G0082	0	0	0	0	0	0	4	0	0	0	0	0	4
145	G0089	0	0	0	0	0	0	2	0	0	0	0	0	2
146	G0100	0	0	0	3	7	0	3	5	3	2	2	0	25
147	G0073	0	0	0	0	0	0	1	0	0	1	0	0	2
148	G0094	0	0	0	0	0	0	1	0	0	2	0	0	3
149	G0097	0	0	0	0	0	0	1	0	0	2	0	0	3
150	G0099	0	0	0	0	0	0	1	0	0	0	0	0	1
151	G0111	0	0	0	0	0	0	0	0	0	2	0	0	2

No	Material Number	<i>Demand</i>												
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	2018 Total
152	G0096	0	0	0	0	0	0	1	0	0	0	0	0	1
153	G0097	0	0	0	0	0	0	1	0	0	0	0	0	1
154	G0089	7	0	7	14	7	0	0	0	0	0	0	0	35
155	G1376	2	0	6	9	27	0	0	0	0	0	0	0	44
156	G1762	10	4	19	6	0	0	0	0	0	0	0	0	39
157	G0373	3	0	0	0	0	0	0	0	0	0	0	0	3

Berikut merupakan data-data atribut tiap *item* yang juga akan digunakan sebagai *input* data.

No.	Material Number	Kode Vendor	Price \$	Satuan	Initial Inventory	Critical Code	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
1	A0116	AD2	28.72	each	1	C	6662.799	2.872	0.5744	15%	33.028	0
2	A0391	AD2	8	each	0	C	195	0.8	0.16	15%	9.2	9
3	A0408	AD2	3.89	each	4	C	558	0.389	0.0778	15%	4.4735	5
4	A0417	AD2	4.36	each	8	C	882	0.436	0.0872	15%	5.014	5
5	A0419	AD2	5.24	each	0	C	882	0.524	0.1048	15%	6.026	0
6	A0510	AD2	2.19	each	0	C	53.28	0.219	0.0438	15%	2.5185	0
7	A0531	AD2	2.88	each	0	C	387.5	0.288	0.0576	15%	3.312	0
8	A0533	AD2	3.38	each	0	B	387.5	0.338	0.0676	30%	4.394	10
9	A0650	AD2	24.02	each	0	C	1188	2.402	0.4804	15%	27.623	2
10	A0668	AD2	11.34	each	0	C	648	1.134	0.2268	15%	13.041	4
11	A0812	AD2	249.57	each	1	C	1505	24.957	4.9914	15%	287.0055	0
12	A1025	AD2	43.76	each	0	C	1505	4.376	0.8752	15%	50.324	0
13	A1216	AD2	190.49	each	4	A	7875	19.049	3.8098	50%	285.735	5
14	A0022	AD2	34.5	each	2	C	144000	3.45	0.69	15%	39.675	0
15	A1857	AD2	314.37	set	0	B	19531.25	31.437	6.2874	30%	408.681	0
16	A1860	AD2	232.47	set	0	B	5120	23.247	4.6494	30%	302.211	2
17	A1876	AD2	299.16	each	1	B	58320	29.916	5.9832	30%	388.908	2
18	A1910	AD2	137	each	1	B	5120	13.7	2.74	30%	178.1	4
19	A2276	AD2	134.29	each	2	B	141354.2	13.429	2.6858	30%	174.577	5
20	A2300	AD2	20.6	each	1	C	32774.13	2.06	0.412	15%	23.69	0
21	A2319	AD2	20.17	each	0	B	3240	2.017	0.4034	30%	26.221	0
22	A2322	AD2	19.94	each	1	C	4050	1.994	0.3988	15%	22.931	0
23	A2330	AD2	64.79	each	2	B	7938	6.479	1.2958	30%	84.227	4
24	A2753	AD2	7.73	each	0	C	11232	0.773	0.1546	15%	8.8895	0
25	A0185	AD2	177	each	1	C	3430	17.7	3.54	15%	203.55	0
26	A0704	AD2	121.41	each	1	C	5022.5	12.141	2.4282	15%	139.6215	0
27	A0705	AD2	153.04	each	0	B	5022.5	15.304	3.0608	30%	198.952	0
28	A0742	AD2	387.51	each	1	B	1640	38.751	7.7502	30%	503.763	0
29	A0753	AD2	241.13	each	2	B	2988.9	24.113	4.8226	30%	313.469	3

No.	Material Number	Kode Vendor	Price \$	Satuan	Initial Inventory	Critical Code	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
30	A0099	AD2	205.3	each	0	C	60800	20.53	4.106	15%	236.095	1
31	A2230	AD2	1.61	each	1	C	24	0.161	0.0322	15%	1.8515	7
32	A2600	AD2	16.09	each	1	C	2.048	1.609	0.3218	15%	18.5035	1
33	A3336	AD2	19.76	each	1	C	1250	1.976	0.3952	15%	22.724	0
34	A4311	AD2	69.93	each	2	A	958.3	6.993	1.3986	50%	104.895	13
35	A0135	AD2	43	each	1	C	250	4.3	0.86	15%	49.45	0
36	A0144	AD2	83.23	each	1	C	360	8.323	1.6646	15%	95.7145	5
37	A0147	AD2	58	each	0	C	360	5.8	1.16	15%	66.7	0
38	A0150	AD2	52.4	each	2	C	640	5.24	1.048	15%	60.26	0
39	A0074	AD2	191.21	eets	0	C	7800	19.121	3.8242	15%	219.8915	0
40	A0120	AD2	28.33	set	2	C	5450	2.833	0.5666	15%	32.5795	4
41	A0134	AD2	57.58	set	0	C	62.6204	5.758	1.1516	15%	66.217	0
42	A0158	AD2	175.64	set	2	C	360	17.564	3.5128	15%	201.986	0
43	A0161	AD2	19.78	set	6	C	360	1.978	0.3956	15%	22.747	6
44	A0162	AD2	13.16	set	1	C	360	1.316	0.2632	15%	15.134	1
45	A0171	AD2	25.22	set	0	C	360	2.522	0.5044	15%	29.003	0
46	A0176	AD2	40.56	set	1	C	360	4.056	0.8112	15%	46.644	2
47	A0060	AD2	419.62	each	2	C	4433	41.962	8.3924	15%	482.563	0
48	A0154	AD2	1.21	each	10	C	25083.82	0.121	0.0242	15%	1.3915	8
49	A0237	AD2	11.67	each	1	C	4860	1.167	0.2334	15%	13.4205	2
50	A0090	AD2	7874.79	each	3	A	56250	787.479	157.4958	50%	11812.19	2
51	A0618	AD2	110.45	each	16	C	352	11.045	2.209	15%	127.0175	0
52	A0934	AD2	49.84	each	19	B	317.52	4.984	0.9968	30%	64.792	0
53	C0818	CI10	515.57	each	0	C	170.1	51.557	10.3114	15%	592.9055	3
54	C0822	CI10	323.42	each	0	C	408.32	32.342	6.4684	15%	371.933	3
55	C0823	CI10	272.13	each	1	C	408.32	27.213	5.4426	15%	312.9495	2
56	C0848	CI10	440.47	each	0	C	6498	44.047	8.8094	15%	506.5405	2
57	C1314	CI10	151.01	each	0	C	29.4	15.101	3.0202	15%	173.6615	0
58	C1362	CI10	13.3	each	1	C	56	1.33	0.266	15%	15.295	2
59	C1364	CI10	2.9	each	6	C	100	0.29	0.058	15%	3.335	6
60	C1366	CI10	4.41	each	0	C	125	0.441	0.0882	15%	5.0715	3

No.	Material Number	Kode Vendor	Price \$	Satuan	Initial Inventory	Critical Code	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
61	C1369	CI10	6.84	each	0	C	56.25	0.684	0.1368	15%	7.866	5
62	C1382	CI10	19.25	each	5	C	4800	1.925	0.385	15%	22.1375	2
63	C0206	CI10	9.75	each	1	A	5475.6	0.975	0.195	50%	14.625	4
64	C0995	CI10	112.18	each	0	C	1054.13	11.218	2.2436	15%	129.007	0
65	C1486	CI10	70.59	each	0	C	201.64	7.059	1.4118	15%	81.1785	0
66	C1488	CI10	110.93	each	0	C	1326.13	11.093	2.2186	15%	127.5695	0
67	C2211	CI10	92.9	each	0	B	1826.72	9.29	1.858	30%	120.77	2
68	C2226	CI10	32.35	each	2	C	30.87	3.235	0.647	15%	37.2025	5
69	C2422	CI10	109.08	each	0	C	5058.054	10.908	2.1816	15%	125.442	0
70	C2425	CI10	126.5	each	0	C	12490.3	12.65	2.53	15%	145.475	0
71	C2503	CI10	154.04	each	0	C	558	15.404	3.0808	15%	177.146	0
72	C0255	CI10	130	each	1	A	216	13	2.6	50%	195	7
73	C0259	CI10	435.55	each	0	B	2499	43.555	8.711	30%	566.215	11
74	C0061	CI10	527.68	each	1	C	33651.55	52.768	10.5536	15%	606.832	0
75	C0002	CI10	1638.11	each	0	A	12000	163.811	32.7622	50%	2457.165	2
76	C1243	CI10	2716.83	each	0	A	2700000	271.683	54.3366	50%	4075.245	3
77	C1945	CI10	1214.37	each	1	C	2304	121.437	24.2874	15%	1396.526	0
78	C2759	CI10	3200	each	0	B	205260	320	64	30%	4160	0
79	C2845	CI10	14.98	each	1	C	205260	1.498	0.2996	15%	17.227	0
80	C2871	CI10	105.03	each	0	B	205260	10.503	2.1006	30%	136.539	0
81	C3079	CI10	438.52	each	2	B	42444	43.852	8.7704	30%	570.076	2
82	C3289	CI10	155.93	each	0	C	38400	15.593	3.1186	15%	179.3195	0
83	C3462	CI10	35.99	each	40	C	260	3.599	0.7198	15%	41.3885	0
84	C3613	CI10	1217.73	each	0	B	3973.12	121.773	24.3546	30%	1583.049	0
85	C3640	CI10	127.8	each	0	B	511.56	12.78	2.556	30%	166.14	0
86	C4119	CI10	654.33	each	1	B	32051.55	65.433	13.0866	30%	850.629	0
87	C0072	CI10	4846.07	each	0	C	2000	484.607	96.9214	15%	5572.981	0
88	C0273	CI10	121.77	each	0	C	449661	12.177	2.4354	15%	140.0355	0
89	C0018	CI10	77.15	each	1	C	16926.56	7.715	1.543	15%	88.7225	1
90	C0075	CI10	516.24	each	6	A	1162.8	51.624	10.3248	50%	774.36	16
91	C0269	CI10	621.35	each	0	B	45.05	62.135	12.427	30%	807.755	0

No.	Material Number	Kode Vendor	Price \$	Satuan	Initial Inventory	Critical Code	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
92	C0281	CI10	44.47	each	1	A	450	4.447	0.8894	50%	66.705	30
93	C0035	CI10	163.12	each	0	C	3990	16.312	3.2624	15%	187.588	0
94	C0074	CI10	177.56	each	0	C	2598.75	17.756	3.5512	15%	204.194	0
95	C0225	CI10	588.64	set	0	C	1350	58.864	11.7728	15%	676.936	0
96	G1203	GE16	41.49	each	0	C	135	4.149	0.8298	15%	47.7135	1
97	G1218	GE16	62.48	each	0	C	30.8	6.248	1.2496	15%	71.852	0
98	G1295	GE16	47.54	each	4	B	1050	4.754	0.9508	30%	61.802	8
99	G1297	GE16	77.45	each	4	A	1890	7.745	1.549	50%	116.175	7
100	G0286	GE16	126.87	each	0	C	667.815	12.687	2.5374	15%	145.9005	0
101	G0472	GE16	11.73	each	0	C	3495.853	1.173	0.2346	15%	13.4895	1
102	G0473	GE16	6.4	set	0	C	4528.718	0.64	0.128	15%	7.36	2
103	G0475	GE16	100.33	each	0	C	15095.73	10.033	2.0066	15%	115.3795	0
104	G0476	GE16	100.33	each	0	C	812.8	10.033	2.0066	15%	115.3795	0
105	G0595	GE16	93.28	each	0	C	1638.4	9.328	1.8656	15%	107.272	0
106	G0596	GE16	245.99	each	0	B	1638.4	24.599	4.9198	30%	319.787	0
107	G0597	GE16	59.12	each	0	C	1638.4	5.912	1.1824	15%	67.988	0
108	G0598	GE16	65.3	each	0	C	1638.4	6.53	1.306	15%	75.095	0
109	G0599	GE16	78.72	each	0	C	1638.4	7.872	1.5744	15%	90.528	0
110	G0732	GE16	163.79	each	0	C	1638.4	16.379	3.2758	15%	188.3585	0
111	G0733	GE16	94.83	each	0	C	1638.4	9.483	1.8966	15%	109.0545	0
112	G0734	GE16	76.01	each	0	B	1638.4	7.601	1.5202	30%	98.813	0
113	G0835	GE16	442.81	each	1	B	360000	44.281	8.8562	30%	575.653	2
114	G1095	GE16	5.87	each	0	C	2127.04	0.587	0.1174	15%	6.7505	2
115	G1878	GE16	205.26	set	0	C	10000	20.526	4.1052	15%	236.049	1
116	G2002	GE16	862.87	each	0	A	686	86.287	17.2574	50%	1294.305	8
117	G2004	GE16	770.03	each	0	A	588	77.003	15.4006	50%	1155.045	8
118	G2343	GE16	230.34	each	0	C	1806	23.034	4.6068	15%	264.891	0
119	G2415	GE16	14.76	each	0	C	4379.2	1.476	0.2952	15%	16.974	1
120	G0288	GE16	72.92	each	1	B	5877.23	7.292	1.4584	30%	94.796	5
121	G0023	GE16	559.02	each	1	B	11220	55.902	11.1804	30%	726.726	2
122	G0062	GE16	522.85	each	1	A	140000	52.285	10.457	50%	784.275	0

No.	Material Number	Kode Vendor	Price \$	Satuan	Initial Inventory	Critical Code	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
123	G0109	GE16	224.88	each	1	A	3429.5	22.488	4.4976	50%	337.32	2
124	G0110	GE16	1865.71	each	1	A	3429.5	186.571	37.3142	50%	2798.565	2
125	G0181	GE16	1606.28	each	0	A	294	160.628	32.1256	50%	2409.42	0
126	G0212	GE16	186.95	set	0	C	2127.04	18.695	3.739	15%	214.9925	1
127	G0213	GE16	236.15	set	0	C	2127.04	23.615	4.723	15%	271.5725	1
128	G0164	GE16	1.37	each	0	B	108	0.137	0.0274	30%	1.781	32
129	G0171	GE16	1.16	each	896	C	576	0.116	0.0232	15%	1.334	0
130	G0765	GE16	13.78	each	0	C	20.16	1.378	0.2756	15%	15.847	2
131	G1043	GE16	40.8	each	1	A	1176	4.08	0.816	50%	61.2	13
132	G2197	GE16	3.63	each	36	C	7.9	0.363	0.0726	15%	4.1745	13
133	G2427	GE16	17.71	each	0	C	1.310965	1.771	0.3542	15%	20.3665	1
134	G2428	GE16	23.62	each	0	C	2.88019	2.362	0.4724	15%	27.163	1
135	G2517	GE16	34.48	each	0	C	76.2	3.448	0.6896	15%	39.652	0
136	G2978	GE16	29.34	set	0	C	688.16	2.934	0.5868	15%	33.741	1
137	G3063	GE16	453.06	each	2	A	288.8	45.306	9.0612	50%	679.59	4
138	G3131	GE16	255.19	each	0	C	1409.805	25.519	5.1038	15%	293.4685	6
139	G3135	GE16	5.86	each	0	C	375.36	0.586	0.1172	15%	6.739	1
140	G3142	GE16	22.12	each	0	B	54	2.212	0.4424	30%	28.756	0
141	G3775	GE16	2.93	each	0	C	1689.12	0.293	0.0586	15%	3.3695	2
142	G4248	GE16	4.92	each	0	C	164.3868	0.492	0.0984	15%	5.658	1
143	G4391	GE16	4.31	each	0	C	164.3868	0.431	0.0862	15%	4.9565	9
144	G0082	GE16	153.45	each	0	C	544.5	15.345	3.069	15%	176.4675	2
145	G0089	GE16	165.72	each	0	C	544.5	16.572	3.3144	15%	190.578	2
146	G0100	GE16	146.17	each	1	A	544.5	14.617	2.9234	50%	219.255	6
147	G0073	GE16	55.6	each	1	C	4913	5.56	1.112	15%	63.94	0
148	G0094	GE16	93.94	each	0	C	1523.61	9.394	1.8788	15%	108.031	0
149	G0097	GE16	32.2	each	0	C	2830.5	3.22	0.644	15%	37.03	0
150	G0099	GE16	91.24	each	0	C	16940	9.124	1.8248	15%	104.926	0
151	G0111	GE16	150.62	each	0	C	16940	15.062	3.0124	15%	173.213	0
152	G0096	GE16	236.15	each	0	C	12500	23.615	4.723	15%	271.5725	1
153	G0097	GE16	245.99	each	0	C	8883.853	24.599	4.9198	15%	282.8885	1

No.	Material Number	Kode Vendor	Price \$	Satuan	<i>Initial Inventory</i>	<i>Critical Code</i>	Volume (cm3)	Holding Cost \$	Minor Order Cost	Penalty	Backorder Cost \$	Safety Stock
154	G0089	GE16	13.02	each	7	C	56000	1.302	0.2604	15%	14.973	5
155	G1376	GE16	20	each	7	C	380	2	0.4	15%	23	5
156	G1762	GE16	9.39	each	20	B	311	0.939	0.1878	30%	12.207	8
157	G0373	GE16	201.19	each	0	C	658800	20.119	4.0238	15%	231.3685	0

(Halaman Ini Sengaja Dikosongkan)

LAMPIRAN C

MODEL LINGO

Pada lampiran berikut merupakan terjemahan model matematis pada Subbab 3.4 yang telah dibangun ke dalam *coding* LINGO.

```
sets:
Item/1..10/:Inv0,volume,PriceItem, CHolding, COrder, CBackLog,
ROL,safety;
Period/1..13/:Carrier;
Linkij(Item,Item):COL,OUTL;
Linkit(Item,Period):demand,LostQ,Inv,OrderQ,RR;
Linkjt(Item,Period):TotalCQ;
Linkijt(Item,Item,Period):kk,qq;
endsets

data:
demand,volume,Inv0,PriceItem,CHolding,COrder,CBackLog,CarrierCap
= @ole ('E:\FALDY\MODEL TA\Data VERIF10x12 3 suppl.xlsx');

@ole ('E:\FALDY\MODEL TA\Data VERIF10x12 3 suppl.xlsx')
= ROL,Carrier,COL,OUTL,OrderQ,Inv,LostQ,RR,kk,qq;

enddata

BigM = 10^7;
StorageCap = 10^10; !Kapasitas Gudang Tak terhingga

TT = @size(period);
!FUNGSI TUJUAN 1
CarrierNumber = @sum(period(t)|t#LT#TT:Carrier(t+1));
!Min = CarrierNumber;

ShortageNumber = @sum(linkit(i,t)|t#LT#TT:LostQ(i,t+1));
!Min = ShortageNumber;

TotalInv = @sum(linkit(i,t)|t#LT#TT:Inv(i,t+1));

!Purchasing Cost;
TotalCPurchase =
@sum(Linkit(i,t)|t#LT#TT:OrderQ(i,t+1)*PriceItem(i));

!Holding Cost;
TotalCHolding = @sum(Linkit(i,t)|t#LT#TT:CHolding(i)*Inv(i,t+1));

!Orderring Cost;
TotalCOrder = @sum(Linkit(i,t)|t#LT#TT:COrder(i)*RR(i,t+1));

!BackLog Cost;
TotalCBackLog =
@sum(Linkit(i,t)|t#LT#TT:CBackLog(i)*LostQ(i,t+1));

!FUNGSI TUJUAN 2;
TotalCost = TotalCPurchase + TotalCHolding + TotalCOrder
+TotalCBackLog;
```

```

!Min = TotalCost ;

!FUNGSI MULTI TUJUAN;
Fitness = ((1-CarrierNumber)/1)^2+((8788.61-TotalCost)/8788.61)^2;
Min = Fitness;

!CONSTRAINT;
!(3.3);
@for (Linkit(i,t) | t#LT#TT: Inv(i,t)+OrderQ(i,t+1) -
Inv(i,t+1)+LostQ(i,t+1)=demand(i,t+1));

!(3.4);
@for
(Period(t) | t#LT#TT: @sum(item(i): volume(i)*OrderQ(i,t+1)) <= CarrierC
ap*Carrier(t+1));

!(3.5);
@for (Linkit(i,t) | t#LT#TT: Inv(i,t) -
demand(i,t+1)+LostQ(i,t+1)+BigM*RR(i,t+1) >= ROL(i));

!(3.6);
@for (Linkit(i,t) | t#LT#TT: Inv(i,t) -demand(i,t+1)+LostQ(i,t+1) -
BigM*(1-RR(i,t+1)) <= ROL(i));

!(3.7);
@for (Linkijt(i,j,t) | t#LT#TT #AND# i#EQ#j: OUTL(i,j) <= Inv(i,t) -
demand(i,t+1)+LostQ(i,t+1)+OrderQ(i,t+1)+BigM*(1-RR(i,t+1)));

!(3.8);
@for (Linkijt(i,j,t) | t#LT#TT #AND# i#EQ#j: OUTL(i,j) >= Inv(i,t) -
demand(i,t+1)+LostQ(i,t+1)+OrderQ(i,t+1)-BigM*(1-RR(i,t+1)));

!(3.9);
@for (Linkij(i,j) | :OUTL(i,j) >= COL(i,j));

!(3.10);
@for (Linkij(i,j) | :COL(i,j) >= ROL(j));

!(3.11);
@for (Linkijt(i,j,t) | #AND# t#LT#TT: Inv(j,t) -
demand(j,t+1)+LostQ(j,t+1)+BigM*kk(i,j,t+1) >= COL(i,j));

!(3.12);
@for (Linkijt(i,j,t) | #AND# t#LT#TT: Inv(j,t) -
demand(j,t+1)+LostQ(j,t+1)-BigM*(1-kk(i,j,t+1)) <= COL(i,j));

!(3.13);
@for (Linkijt(i,j,t) | #AND# t#LT#TT: qq(i,j,t+1) <= kk(i,j,t+1));

!(3.14);
@for (Linkijt(i,j,t) | #AND# t#LT#TT: qq(i,j,t+1) <= RR(i,t+1));

!(3.15);
@for (Linkjt(j,t) | t#LT#TT: @sum(item(i): qq(i,j,t+1)) <= 1);

!(3.16);

```

```

@for (Linkjt(j,t) | t#LT#TT:TotalCQ(j,t) =
@sum(item(i):COL(i,j)*qq(i,j,t+1)));

@for (Linkijt(i,j,t) | #AND#
t#LT#TT:COL(i,j)*kk(i,j,t+1)<=TotalCQ(j,t));

!(3.17);
@for (Linkijt(i,j,t) | #AND# t#LT#TT:OUTL(i,j)<=Inv(j,t)-
demand(j,t+1)+LostQ(j,t+1)+OrderQ(j,t+1)+BigM*(1-qq(i,j,t+1)));

!(3.18);
@for (Linkijt(i,j,t) | #AND# t#LT#TT:OUTL(i,j)>=Inv(j,t)-
demand(j,t+1)+LostQ(j,t+1)+OrderQ(j,t+1)-BigM*(1-qq(i,j,t+1)));

!(3.19);
@for
(Linkjt(j,t) | t#LT#TT:OrderQ(j,t+1)<=BigM*(RR(j,t+1)+@sum(item(i):q
q(i,j,t+1))));
!(3.20);
@for (Linkit(i,t) | t#LT#TT:LostQ(i,t+1)<=demand(i,t+1));

!(3.21) Kapasitas Gudang;
@for (Period(t) | t#LT#TT:@sum(item(i): OrderQ(i,t+1))<=StorageCap;

!Inventory Awal;
@for (Linkit(i,t) | t#EQ#1:Inv(i,t)=Inv0(i));

@for (item(j):ROL(j)>=safety(j));

!Maximum Stockout probability;
@for (item(i):@sum(period(t) | t#LT#TT:LostQ(i,t+1)) <=
@sum(period(t) | t#LT#TT:demand(i,t+1)));

!Binary & Integer Variabel;
@for (Linkijt(i,j,t):@bin(kk(i,j,t));@bin(qq(i,j,t)));
@for
(Linkit(i,t):@bin(RR(i,t));@gin(LostQ(i,t));@GIN(Inv(i,t));@GIN(Or
derQ(i,t)));
@for (Linkij(i,j):@gin(COL(i,j));@gin(OUTL(i,j)));
@for (Period(t):@GIN(Carrier(t)));
@for (Item(i):@GIN(ROL(i)));

```

(Halaman Ini Sengaja Dikosongkan)

LAMPIRAN D

CODING MATLAB

Kode Program untuk Membangkitkan Solusi Random

```
function [s, c, S, o, Inv, x, r, k, q, Z1, Z2] = FMCOLCOPRandom
(demand, Inv0, price, CH, CO, CB, v, cap, supplier, safetystock)
[I, T]=size(demand);
J=I;
Inv=[Inv0 zeros(I, T-1)-1];
o=zeros(I, T);
x=zeros(I, T);
r=zeros(I, T);
k=zeros(I, J, T);
q=zeros(I, J, T);
s=zeros(I, 1);
c=zeros(I, J);
S=zeros(I, J);
maxdemand=max(demand');
mindemand=min(demand');
totaldemand=sum(demand, 2);

%menghitung banyaknya supplier
nsupplier=max(supplier)-min(supplier)+1;
supawal=min(supplier);
supakhir=max(supplier);

for i=1:nsupplier
    idxsup{i}=find(supplier==i);
end

%membangkitkan solusi reorder level
for i = 1:I
    slower = safetystock(i);
    supper = round(1*safetystock(i));
    s(i)=floor(rand()*(supper-slower) + slower);
end

%membangkitkan solusi can order level
for i = 1:I
    for j = 1:J
        clower = s(j);
        cupper = round(1.5*s(j));
        if i==j
            c(i, j)=s(j);
        else
            c(i, j)=floor(rand()*(cupper-clower) + clower);
        end
    end
end

%membangkitkan solusi order up to level
for i = 1:I
    for j = 1:J
        Slower = c(i, j);
        Supper = totaldemand(j)-Inv0(j);
        S(i, j)=floor(rand()*(Supper-Slower) + Slower);
    end
end
```

```

        end
    end

    %Membuat nilai item yang tidak terkorelasi menjadi 0
    for i=1:nsupplier
        for j=1:nsupplier
            for a=1:length(idxsup{i})
                for b=1:length(idxsup{j})
                    if i~=j
                        c(idxsup{i}(a),idxsup{j}(b))=0;
                        S(idxsup{i}(a),idxsup{j}(b))=0;
                    end
                end
            end
        end
    end

    %menentukan hasil output berdasarkan konstrain dan dec. variabel
    for t = 1:T-1
        for i = 1:I
            for j=1:J
                if Inv(i,t)-demand(i,t+1)+o(i,t+1)>s(i)
                    r(i,t+1)=0;
                else
                    r(i,t+1)=1;
                end
                if r(i,t+1)==1
                    x(i,t+1)=min(max(S(i,i)-(Inv(i,t)-
demand(i,t+1)+o(i,t+1)),0),sum(demand(i,t+1:T))+safetystock(i));
                else
                    x(i,t+1)=0;
                end

                if Inv(j,t)-demand(j,t+1)+o(j,t+1)>c(i,j) && Inv(j,t)-
demand(j,t+1)+o(j,t+1)<=s(j)
                    k(i,j,t+1)=0;

                else
                    k(i,j,t+1)=1;
                end

                if r(j,t+1)==0 && k(i,j,t+1)==0
                    q(i,j,t+1)=0;
                    x(j,t+1)=0;
                else
                    random=rand();
                    if random>rand()
                        q(:,j,t+1)=zeros(1,I);
                    else
                        q(:,j,t+1)=(randperm(I)==I);
                    end
                end

                if q(i,j,t+1)==1 && i ~= j
                    x(j,t+1)=min(max(S(i,j)-(Inv(j,t)-
demand(j,t+1)+o(j,t+1)),0),sum(demand(j,t+1:T))+safetystock(j));

```



```

        else
            q(i,j,t+1)=0;
        end
        Inv(j,t+1)=Inv(j,t)+x(j,t+1)+o(j,t+1)-demand(j,t+1);
    end
end
end

%menentukan besarnya item stockout
[ii tt]=find(Inv<0);
o(ii,tt) = -Inv(ii,tt);
Inv(ii,tt)=0;

ix=find(x==0);
r(ix)=0;
orderc=zeros(nsupplier,T);
for i = 1: nsupplier
    ncarrier(i)=sum(ceil(sum(x(idxsup{i},:).* repmat(v(idxsup{i},:),1,T)
    )/cap));
    orderc(i,:)=sum(x(idxsup{i},:).* repmat(price(idxsup{i},:),1,T));
end

%menghitung major orderring cost
TMOC=0;
for i=1:nsupplier
    for t=1:T
        if orderc(i,t) <769.23
            MOC=38.46;
        if orderc(i,t) >= 7692.3
            MOC=153.85;
        else
            MOC=76.92;
        end
    end
    TMOC=TMOC+MOC;
end
end

%Fungsi Tujuan 2
Z2=TMOC+sum(sum(x.*repmat(price,1,T)+r.*repmat(CO,1,T)+o.*repmat(C
B,1,T)))+sum(sum(Inv(:,2:T).* repmat(CH,1,T-1)));

%Fungsi Tujuan 1
Z1=sum(ncarrier);

```

Kode Program untuk Mencari Nilai Solusi Terdekat

```
function [snew, cnew, Snew, onew, Invnew, xnew, rnew, knew, qnew, Z1, Z2] = FNeighborSolution(demand, Inv0, price, CH, CO, CB, v, cap, s, c, S, o, supplier, safetystock)

[I, T] = size(demand);
J = I;
Invnew = [Inv0 zeros(I, T-1) -1];
onew = zeros(I, T);
xnew = zeros(I, T);
rnew = zeros(I, T);
knew = zeros(I, J, T);
qnew = zeros(I, J, T);
snew = zeros(I, 1);
cnew = zeros(I, J);
Snew = zeros(I, J);
maxdemand = max(demand');
mindemand = min(demand');
totaldemand = sum(demand, 2);
nsupplier = max(supplier) - min(supplier) + 1;
supawal = min(supplier);
supakhir = max(supplier);

for i = 1:nsupplier
    idxsup{i} = find(supplier == i);
end

%menentukan besarnya penyimpangan perubahan sebesar 1
cge = ones(I, 1);

% Mencari solusi reorder level dari solusi yang telah dibangkitkan
for i = 1:I
    supper = round(1 * safetystock(i));
    snew(i) = min(max(s(i) + round(2 * cge(i) * rand() - cge(i)), safetystock(i)), supper);
end

% Mencari solusi can order level dari solusi yang telah dibangkitkan
for i = 1:I
    for j = 1:J
        cupper = round(1.2 * s(j));

        if i == j
            cnew(i, j) = snew(j);
        else
            cnew(i, j) = min(max(c(i, j) + round(2 * cge(j) * rand() - cge(j)), snew(j)), cupper);
        end
    end
end

% Mencari solusi order up to level dari solusi yang telah dibangkitkan
for i = 1:I
    for j = 1:J
```

```

        Slower = cnew(i,j);
        Supper = totaldemand(j)-Inv0(j);
        Snew(i,j)=min(max(S(i,j)+round(2*cge(j)*rand()-
cge(j)),Slower),Supper);
    end
end

%Membuat nilai item yang tidak terkorelasi menjadi 0
for i=1:nsupplier
    for j=1:nsupplier
        for a=1:length(idxsup{i})
            for b=1:length(idxsup{j})
                if i~=j
                    cnew(idxsup{i}(a),idxsup{j}(b))=0;
                    Snew(idxsup{i}(a),idxsup{j}(b))=0;
                end
            end
        end
    end
end

%menentukan hasil output berdasarkan konstrain dan dec. variabel
for t = 1:T-1
    for i = 1:I
        for j=1:J
            if Invnew(i,t)-demand(i,t+1)+onew(i,t+1)>snew(i)
                rnew(i,t+1)=0;
            else
                rnew(i,t+1)=1;
            end
            if rnew(i,t+1)==1
                xnew(i,t+1)=min(max(Snew(i,i)-(Invnew(i,t)-
demand(i,t+1)+onew(i,t+1)),0),sum(demand(i,t+1:T))+safetystock(i))
;
            else
                xnew(i,t+1)=0;
            end

            if Invnew(j,t)-demand(j,t+1)+onew(j,t+1)>cnew(i,j)
                knew(i,j,t+1)=0;
            else
                knew(i,j,t+1)=1;
            end

            if rnew(j,t+1)==0 && knew(i,j,t+1)==0
                qnew(i,j,t+1)=0;
                xnew(j,t+1)=0;
            else
                random=rand();
                if random>rand()
                    qnew(:,j,t+1)=zeros(1,I);
                else
                    qnew(:,j,t+1)=(randperm(I)==I);
                end
            end

            if qnew(i,j,t+1)==1 && i ~= j

```

```

        xnew(j,t+1)=min(max(Snew(i,j)-(Invnew(j,t)-
demand(j,t+1)+onew(j,t+1)),0),sum(demand(j,t+1:T)+safetystock(j)))
;

        else
            qnew(i,j,t+1)=0;
        end
        Invnew(i,t+1)=Invnew(i,t)+xnew(i,t+1)+onew(i,t+1)-
demand(i,t+1);
        Invnew(j,t+1)=Invnew(j,t)+xnew(j,t+1)+onew(j,t+1)-
demand(j,t+1);

    end
end
end

%menentukan besarnya item stockout
[ii tt]=find(Invnew<0);
onew(ii,tt) = -Invnew(ii,tt);
Invnew(ii,tt)=0;

ix=find(xnew==0);
rnew(ix)=0;

orderc=zeros(nsupplier,T);
for i = 1: nsupplier
    ncarrier(i)=sum(ceil(sum(xnew(idxsup{i},:).*repmat(v(idxsup{i},:),
1,T))/cap));
    orderc(i,:)=sum(xnew(idxsup{i},:).*repmat(price(idxsup{i},:),1,T))
;
end

%menghitung major orderring cost
TMOC=0;

for i=1:nsupplier
    for t=1:T
        if orderc(i,t) <769.23
            MOC=38.46;
        if orderc(i,t) >= 7692.3
            MOC=153.85;
        else
            MOC=76.92;
        end
    end
    TMOC=TMOC+MOC;
end
end

%Fungsi Tujuan 2
Z2=sum(sum(xnew.*repmat(price,1,T)+rnew.*repmat(CO,1,T)+onew.*repm
at(CB,1,T)))+sum(sum(Invnew(:,2:T).*repmat(CH,1,T-1)));

%Fungsi Tujuan 1
Z1=sum(ncarrier);

```

Kode Program untuk Mencari Fungsi Tujuan 1 Menggunakan Algoritma Simulated Annealing

```

function [sbest, cbest, Sbest, obest, Invbest, xbest, rbest,
kbest,
qbest,waktukomputasi,fbest2,fbest1]=FSAMCOLZ1(demand,Inv0,price,CH
,CO,CB,v,cap,supplier,safetystock,maxiter,cr)
waktukomputasi = cputime;
fbesta=[];

%membangkitkan solusi random
[s, c, S, o, Inv, x, r, k, q, Z1, Z2] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);
for n=1:10

[s(:,n),c(:,n),S(:,n),o(:,n),Inv(:,n),x(:,n),r(:,n),k(
:,:,n),q(:,:,n),Z1(n),Z2(n)] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);
    Z1(n);
End

T=mean(Z1);%menentukan suhu awal dengan rata2
[minZ,idx]=min(Z1);
s=s(:,idx);c=c(:,idx);S=S(:,idx);o=o(:,idx);Inv=Inv(:,idx)
;x=x(:,idx);r=r(:,idx);k=k(:,idx);q=q(:,idx);
fbest1=minZ; %memilih solusi terbaik dari hasil random yang
dibangkitkan
it=1;

%Algoritma Simulated Annealing
while T>=10^-4
snew=s;cnew=c;Snew=S;onew=o;Invnew=Inv;xnew=x;rnew=r;knew=k;qnew=q
;
[snew, cnew, Snew, onew, Invnew, xnew, rnew, knew, qnew, Z1,
Z2new] = FNeighborSolution
(demand,Inv0,price,CH,CO,CB,v,cap,s,c,S,o,supplier,safetystock);
    if Z1<fbest1
        fbest1=Z1;

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z2=Z2new;
    else
        deltaf=Z1-fbest1;
        PE=exp(-deltaf/T); %menghitung probabilitas boltzmann
        if rand<PE
            fbest1=Z1;

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z2=Z2new;
        end
    end
    it=it+1;
    if it == maxiter
        T=cr*T; %mereduksi temperatur bila mencapai batas
siklus
        it=1;
    end
    fbesta=[fbesta;fbest1];

```

```
end

sbest=s;cbest=c;Sbest=S;obest=o;Invbest=Inv;xbest=x;rbest=r;kbest=
k;qbest=q;fbest2=Z2;
    fbest1=fbest1;

waktukomputasi=cputime-waktukomputasi;
plot(fbesta);
end
```

Kode Program untuk Mencari Fungsi Tujuan 2 Menggunakan Algoritma Simulated Annealing

```
function [sbest, cbest, Sbest, obest, Invbest, xbest, rbest,
kbest,
qbest,waktukomputasi,fbest1,fbest2]=FSAMCOLZ2(demand,Inv0,price,CH
,CO,CB,v,cap,supplier,safetystock,maxiter,cr)
waktukomputasi = cputime;
fbesta=[];

%membangkitkan solusi random
[s, c, S, o, Inv, x, r, k, q, Z1, Z2] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);
for n=1:10

[s(:,n),c(:,n),S(:,n),o(:,n),Inv(:,n),x(:,n),r(:,n),k(
:,:,:),q(:,n),Z1(n),Z2(n)] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);
Z2(n);
end
T=mean(Z2);%menentukan suhu awal dengan rata2
[minZ,idx]=min(Z2);
s=s(:,idx);c=c(:,idx);S=S(:,idx);o=o(:,idx);Inv=Inv(:,idx)
;x=x(:,idx);r=r(:,idx);k=k(:,idx);q=q(:,idx);
fbest2=minZ; %memilih solusi terbaik dari hasil random yang
dibangkitkan
it=1;

%Algoritma SImulated Annealing
while T>=10^-4

snew=s;cnew=c;Snew=S;onew=o;Invnew=Inv;xnew=x;rnew=r;knew=k;qnew=q
;
[snew, cnew, Snew, onew, Invnew, xnew, rnew, knew, qnew, Z1new,
Z2] = FNeighborSolution
(demand,Inv0,price,CH,CO,CB,v,cap,s,c,S,o,supplier,safetystock);
if Z2<fbest2
fbest2=Z2;

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z1=Z1new;
else
deltaf=Z2-fbest2;
PE=exp(-deltaf/T); %menghitung probabilitas boltzmann
if rand<PE
fbest2=Z2;

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z1=Z1new;
end
end
it=it+1;
if it == maxiter
T=cr*T; %mereduksi temperatur bila mencapai batas
siklus
it=1;
end
fbesta=[fbesta;fbest2];
```

```
end

sbest=s;cbest=c;Sbest=S;obest=o;Invbest=Inv;xbest=x;rbest=r;kbest=
k;qbest=q;fbest1=Z1;
    fbest2=fbest2;

waktukomputasi=cputime-waktukomputasi;
plot(fbesta);
end
```


Kode Program untuk Mencari Fungsi Multi Objective Menggunakan Algoritma Simulated Annealing

```

function [sbest, cbest, Sbest, obest, Invbest, xbest, rbest,
kbest,
qbest,waktukomputasi,Z1opt,Z2opt,fbest]=FSAMCOLMULTI(demand,Inv0,p
rice,CH,CO,CB,v,cap,supplier,safetystock,maxiter,cr)
waktukomputasi = cputime;
fbesta=[];

%menghitung fungsi tujuan 1
[sbest, cbest, Sbest, obest, Invbest, xbest, rbest, kbest,
qbest,waktukomputasi,fbest2,fbestlopt]=FSAMCOLZ1(demand,Inv0,price
,CH,CO,CB,v,cap,supplier,safetystock,maxiter,cr);

%menghitung fungsi tujuan 1
[sbest, cbest, Sbest, obest, Invbest, xbest, rbest, kbest,
qbest,waktukomputasi,fbest1,fbest2opt]=FSAMCOLZ2(demand,Inv0,price
,CH,CO,CB,v,cap,supplier,safetystock,maxiter,cr);

fbestlopt=fbestlopt;
fbest2opt=fbest2opt;

%membangkitkan solusi random
[s, c, S, o, Inv, x, r, k, q, Z1, Z2] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);

for n=1:10

[s(:,n),c(:,n),S(:,n),o(:,n),Inv(:,n),x(:,n),r(:,n),k(
(:,n),q(:,n),Z1(n),Z2(n)] = FMCOLCOPRandom
(demand,Inv0,price,CH,CO,CB,v,cap,supplier,safetystock);
    fbest(n) = ((fbestlopt-Z1(n))/fbestlopt)^2 + ((fbest2opt-
Z2(n))/fbest2opt)^2;
end

T=mean(fbest);%menentukan suhu awal dengan rata2
[minZ,idx]=min(fbest);
s=s(:,idx);c=c(:,idx);S=S(:,idx);o=o(:,idx);Inv=Inv(:,idx)
;x=x(:,idx);r=r(:,idx);k=k(:,idx);q=q(:,idx);
fbest=minZ; %memilih solusi terbaik dari hasil random yang
dibangkitkan
it=1;

%Algoritma Simulated Annealing
while T>=10^-4

snew=s;cnew=c;Snew=S;onew=o;Invnew=Inv;xnew=x;rnew=r;knew=k;qnew=q
;Z1new=Z1;Z2new=Z2;
[snew, cnew, Snew, onew, Invnew, xnew, rnew, knew, qnew, Z1new,
Z2new] = FNeighborSolution
(demand,Inv0,price,CH,CO,CB,v,cap,s,c,S,o,supplier,safetystock);
fglobal=((fbestlopt-Z1new)/fbestlopt)^2+((fbest2opt-
Z2new)/fbest2opt)^2;
    if fglobal<fbest
        fbest=fglobal;
    
```

```

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z1=Z1new;Z2=Z2new;
    else
        deltaf=fglobal-fbest;
        PE=exp(-deltaf/T); %menghitung probabilitas boltzmann
        if rand<PE
            fbest=fglobal;

s=snew;c=cnew;S=Snew;o=onew;Inv=Invnew;x=xnew;r=rnew;k=knew;q=qnew
;Z1=Z1new;Z2=Z2new;
    end
    end
    it=it+1;
    if it == maxiter
        T=cr*T; %mereduksi temperatur bila mencapai batas
siklus
        it=1;
    end
    fbesta=[fbesta;fbest];
end

sbest=s;cbest=c;Sbest=S;obest=o;Invbest=Inv;xbest=x;rbest=r;kbest=
k;qbest=q;Z1opt=Z1;Z2opt=Z2;
    fbest=fbest;

waktukomputasi=cputime-waktukomputasi;
plot(fbesta);
end

```

LAMPIRAN E

PERHITUNGAN KONDISI EKSISTING

Berikut merupakan skema pembelian dari kebijakan persediaan yang diterapkan perusahaan.

Jumlah Pemesanan Kondisi Eksisting

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0116	13	0	0	0	0	0	13	0	0	0	0	0
A0391	9	0	0	0	0	0	24	0	0	0	0	0
A0408	9	0	0	1	0	0	2	0	0	0	0	0
A0417	0	0	48	0	0	0	1	0	0	16	0	0
A0419	0	0	0	0	0	0	1	0	0	0	0	0
A0510	0	0	0	1	0	0	0	0	0	0	0	0
A0531	0	0	0	2	0	0	0	0	0	0	0	0
A0533	10	0	0	0	0	0	24	0	0	0	0	0
A0650	6	0	0	5	0	0	0	0	0	0	0	0
A0668	4	0	0	4	0	0	0	0	0	0	0	0
A0812	0	0	0	0	0	0	1	0	0	0	0	0
A1025	2	0	0	2	0	0	4	0	0	0	0	0
A1216	9	0	0	4	0	0	0	0	0	0	3	0
A0022	6	0	0	0	0	0	2	0	0	0	0	0
A1857	0	0	0	4	0	0	0	0	0	0	0	0
A1860	7	0	0	0	7	0	0	0	0	0	7	0
A1876	1	0	0	2	0	0	0	0	0	0	0	0
A1910	12	0	0	20	0	0	2	0	0	20	0	0
A2276	9	0	0	6	0	0	2	0	0	0	6	0
A2300	0	0	0	0	0	0	1	0	0	0	0	0
A2319	0	0	0	2	0	0	2	0	0	0	0	0
A2322	0	3	0	0	0	0	1	0	0	0	0	0
A2330	2	0	0	2	0	0	2	0	0	0	0	0
A2753	0	0	0	4	0	0	0	0	0	0	0	0
A0185	14	0	0	0	0	0	14	0	0	0	0	0
A0704	1	0	0	0	0	0	0	0	0	0	0	0
A0705	3	0	0	0	3	0	0	0	0	0	0	0
A0742	1	0	0	0	1	0	0	0	0	0	0	0
A0753	5	0	0	0	5	0	0	0	0	0	0	0
A0099	2	0	0	1	0	0	0	0	0	0	0	0
A2230	6	0	0	9	0	0	16	0	0	8	0	0
A2600	1	0	0	8	0	0	0	0	0	0	0	0
A3336	0	0	0	0	0	0	1	0	0	0	0	0
A4311	0	115	0	42	0	0	16	0	0	0	0	0
A0135	1	0	0	0	0	0	0	0	0	0	0	0
A0144	15	0	0	60	0	0	0	0	20	0	0	0
A0147	2	0	0	0	2	0	0	0	0	2	0	0
A0150	2	0	0	0	0	0	2	0	0	0	0	0
A0074	0	0	0	1	0	0	2	0	0	1	0	0
A0120	2	0	0	2	0	0	4	0	0	2	0	0
A0134	1	0	0	0	0	0	0	0	0	0	0	0
A0158	1	0	0	0	0	0	0	0	0	0	0	0
A0161	4	0	0	11	0	0	0	0	0	0	0	0
A0162	1	0	0	1	0	0	0	0	0	0	0	0
A0171	0	0	0	1	0	0	0	0	0	0	0	0
A0176	5	0	0	0	0	0	0	0	0	0	0	0
A0060	2	0	0	0	0	0	0	0	0	0	0	0
A0154	14	0	0	12	0	0	0	0	0	0	0	0
A0237	5	0	0	0	0	0	0	0	0	0	0	0

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0090	1	0	0	0	0	0	2	0	0	1	0	0
A0618	16	0	0	0	0	0	16	0	0	0	0	0
A0934	0	0	0	0	0	0	19	0	0	0	0	0
C0818	3	0	0	4	0	0	3	0	0	0	0	0
C0822	4	0	0	6	0	0	0	0	0	0	0	0
C0823	5	0	0	3	0	0	0	0	0	1	0	0
C0848	2	0	0	2	0	0	2	0	0	0	0	0
C1314	0	0	0	0	0	0	4	0	0	0	0	0
C1362	1	0	0	2	0	0	0	0	0	0	0	0
C1364	0	0	16	0	0	0	0	0	16	0	0	0
C1366	3	0	0	4	0	0	0	0	0	0	0	0
C1369	5	0	0	8	0	0	0	0	0	0	0	0
C1382	0	0	0	1	0	0	1	0	0	1	0	0
C0206	32	0	0	9	0	0	0	0	0	0	0	0
C0995	1	0	0	0	0	0	0	0	0	0	0	0
C1486	1	0	0	0	0	0	0	0	0	0	0	0
C1488	1	0	0	0	0	0	0	0	0	0	0	0
C2211	2	0	0	4	0	0	0	0	0	0	0	0
C2226	19	0	0	0	0	0	0	0	0	0	0	0
C2422	1	0	0	0	0	0	0	0	0	0	0	0
C2425	1	0	0	1	0	0	0	0	0	0	0	0
C2503	2	0	0	0	0	0	0	0	0	0	0	0
C0255	26	0	0	0	0	26	0	0	26	0	0	0
C0259	11	0	0	76	0	0	0	0	21	0	0	0
C0061	2	0	0	0	0	0	0	0	0	0	0	0
C0002	2	0	0	2	0	0	2	0	0	1	0	0
C1243	3	0	0	2	0	0	2	0	0	1	0	0
C1945	0	0	0	0	0	0	1	0	0	0	0	0
C2759	0	0	0	1	0	0	0	0	0	0	0	0
C2845	0	0	0	0	0	0	23	0	0	0	0	0
C2871	4	0	0	8	0	0	0	0	0	0	0	0
C3079	6	0	0	0	0	0	0	0	0	0	0	0
C3289	0	0	0	2	0	0	0	0	0	0	0	0
C3462	0	0	0	0	0	0	40	0	0	0	0	0
C3613	1	0	0	0	0	0	0	0	0	0	0	0
C3640	15	0	0	30	0	0	0	0	0	0	0	0
C4119	1	0	0	8	0	0	0	0	0	0	0	0
C0072	1	0	0	0	0	0	0	0	0	0	0	0
C0273	9	0	0	6	0	0	0	0	0	0	0	0
C0018	2	0	0	1	0	0	0	0	0	0	0	0
C0075	26	0	0	16	0	0	8	0	0	4	0	0
C0269	9	0	0	0	0	0	0	0	0	0	0	0
C0281	36 9	0	0	0	0	0	312	0	0	0	0	0
C0035	1	0	0	0	0	0	0	0	0	0	0	0
C0074	1	0	0	0	0	0	0	0	0	0	0	0
C0225	0	0	0	0	0	0	1	0	0	0	0	0
G1203	1	0	0	0	0	0	1	0	0	0	0	0
G1218	0	0	0	0	0	0	4	0	0	0	0	0
G1295	4	0	0	8	0	0	8	0	0	4	0	0
G1297	3	0	0	8	0	0	8	0	0	4	0	0
G0286	0	0	0	1	0	0	0	0	0	0	0	0
G0472	1	0	0	0	0	0	1	0	0	0	0	0
G0473	2	0	0	0	0	0	2	0	0	0	0	0
G0475	0	0	0	2	0	0	0	0	0	0	0	0
G0476	0	0	0	1	0	0	0	0	0	0	0	0
G0595	0	0	0	1	0	0	0	0	0	0	0	0
G0596	0	0	0	1	0	0	0	0	0	0	0	0
G0597	0	0	0	1	0	0	0	0	0	0	0	0

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
G0598	0	0	0	1	0	0	0	0	0	0	0	0
G0599	0	0	0	2	0	0	0	0	0	0	0	0
G0732	0	0	0	1	0	0	0	0	0	0	0	0
G0733	0	0	0	1	0	0	0	0	0	0	0	0
G0734	0	0	0	4	0	0	0	0	0	0	0	0
G0835	3	0	0	0	0	0	0	0	0	0	0	0
G1095	2	0	0	0	0	0	2	0	0	0	0	0
G1878	1	0	0	1	0	0	0	0	0	0	0	0
G2002	8	0	0	12	0	0	0	0	12	0	0	0
G2004	8	0	0	12	0	0	0	0	12	0	0	0
G2343	0	0	0	1	0	0	0	0	0	0	0	0
G2415	1	0	0	0	0	0	1	0	0	0	0	0
G0288	4	0	0	10	0	0	11	0	0	4	0	0
G0023	1	0	0	0	0	0	2	0	0	0	0	0
G0062	0	0	0	1	0	0	0	0	0	0	0	0
G0109	1	0	0	2	0	0	2	0	0	1	0	0
G0110	1	0	0	1	0	0	2	0	0	1	0	0
G0181	0	0	0	3	0	0	0	0	0	0	0	0
G0212	1	0	0	0	0	0	1	0	0	0	0	0
G0213	1	0	0	0	0	0	1	0	0	0	0	0
G0164	78 4	0	0	0	0	0	120	0	0	0	0	0
G0171	22 0	0	0	0	0	55	0	0	0	0	0	0
G0765	2	0	0	0	0	0	2	0	0	0	0	0
G1043	40	0	0	40	0	0	6	0	0	3	0	0
G2197	12 1	0	0	0	121	0	0	0	0	0	0	0
G2427	1	0	0	0	0	0	1	0	0	0	0	0
G2428	1	0	0	0	0	0	1	0	0	0	0	0
G2517	0	0	0	2	0	0	0	0	0	0	0	0
G2978	1	0	0	0	0	0	1	0	0	0	0	0
G3063	2	0	0	8	0	0	0	0	0	0	0	0
G3131	6	0	0	12	0	0	0	0	0	0	0	0
G3135	1	0	0	0	0	0	1	0	0	0	0	0
G3142	0	0	0	0	0	0	4	0	0	0	0	0
G3775	2	0	0	0	0	0	2	0	0	0	0	0
G4248	1	0	0	0	0	0	1	0	0	0	0	0
G4391	9	0	0	18	0	0	0	0	0	0	0	0
G0082	2	0	0	0	0	0	4	0	0	0	0	0
G0089	2	0	0	0	0	0	2	0	0	0	0	0
G0100	5	0	0	10	0	0	11	0	0	10	0	0
G0073	0	0	0	0	0	0	0	0	0	1	0	0
G0094	0	0	0	0	0	0	3	0	0	2	0	0
G0097	0	0	0	0	0	0	1	0	0	2	0	0
G0099	0	0	0	0	0	0	1	0	0	0	0	0
G0111	0	0	0	0	0	0	0	0	0	2	0	0
G0096	1	0	0	0	0	0	1	0	0	0	0	0
G0097	1	0	0	0	0	0	1	0	0	0	0	0
G0089	12	0	0	21	0	0	0	0	0	0	0	0
G1376	6	0	0	36	0	0	0	0	0	0	0	0
G1762	21	0	0	6	0	0	21	0	0	0	0	0
G0373	9	0	3	0	0	3	0	0	3	0	0	3
Volume A	3163771.6	122354.5	42336	1704269.2	68212	0	978961.55	0	7200	192374	907590.05	0

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Volume C	13848488	0	1600	10635612	0	5616	10327160	0	59695	2721859.5	0	0
Volume G	8088075.4	0	1976400	1578260.1	955.9	2008080	236421.72	0	1991688	98602.14	0	1976400
Carrier Sup A	4	1	1	2	1	0	1	0	1	1	1	0
Carrier Sup C	14	0	1	11	0	1	10	0	1	3	0	0
Carrier Sup G	9	0	2	2	1	2	1	0	2	1	0	2
Biaya Pembelian \$	130412.94	8101.77	859.25	128136.19	4234.8	4047.37	70086.77	0	34435.92	22848.03	3004.5	603.57
Ordering Cost \$	7488.1269	118.08072	179.33042	5164.5465	378.42725	183.04742	4698.4468	0	478.41925	1814.8938	189.79442	63.728008

Jumlah Pemesanan Kondisi Eksisting

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0116	14	0	0	0	0	0	13	13	13	13	13	13
A0391	9	9	9	9	9	9	9	9	9	9	9	9
A0408	5	5	5	5	5	5	5	5	5	5	5	5
A0417	8	8	56	7	6	5	5	5	5	5	5	5
A0419	0	0	0	0	0	0	1	1	0	0	0	0
A0510	0	0	0	0	0	0	0	0	0	0	0	0
A0531	0	0	0	2	2	0	0	0	0	0	0	0
A0533	10	10	10	10	10	10	10	10	10	10	10	10
A0650	3	2	2	4	2	2	2	2	2	2	2	2
A0668	4	4	4	8	8	4	4	4	4	4	4	4
A0812	1	1	1	0	0	0	0	0	0	0	0	0
A1025	2	0	0	2	0	0	2	0	0	0	0	0
A1216	5	5	5	5	5	5	5	5	5	5	8	8
A0022	8	8	8	6	6	0	0	0	0	0	0	0
A1857	0	0	0	0	0	0	0	0	0	0	0	0
A1860	3	3	2	2	9	9	9	9	9	9	16	16
A1876	2	2	2	2	2	2	2	2	2	2	2	2
A1910	12	12	4	12	9	4	6	5	4	24	24	24
A2276	11	11	5	9	5	5	7	5	5	5	11	11
A2300	1	1	1	1	1	0	1	1	0	0	0	0
A2319	0	0	0	2	2	0	2	2	0	0	0	0
A2322	1	4	4	4	3	3	4	3	3	3	3	3
A2330	4	4	4	6	4	4	6	4	4	4	4	4
A2753	0	0	0	4	4	0	0	0	0	0	0	0
A0185	0	0	0	0	0	0	14	14	14	14	14	14
A0704	1	1	0	0	0	0	0	0	0	0	0	0
A0705	0	0	0	0	3	3	3	3	3	3	3	3

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0742	1	1	0	0	1	1	1	1	1	1	1	1
A0753	5	5	3	3	8	8	8	8	8	8	8	8
A0099	1	1	1	1	1	1	1	1	1	1	1	1
A2230	7	7	7	15	15	7	7	7	7	15	7	7
A2600	1	1	1	1	1	1	1	1	1	1	1	1
A3336	1	1	1	1	0	0	1	0	0	0	0	0
A4311	74	27	13	41	27	13	13	13	13	13	13	13
A0135	2	0	0	0	0	0	0	0	0	0	0	0
A0144	16	16	5	65	5	5	5	5	25	25	25	25
A0147	2	0	0	0	2	2	2	2	2	4	4	4
A0150	4	0	0	0	0	0	2	2	2	2	2	2
A0074	0	0	0	1	1	0	2	2	2	2	2	2
A0120	4	4	4	6	6	4	4	4	4	6	4	4
A0134	1	0	0	0	0	0	0	0	0	0	0	0
A0158	2	2	2	0	0	0	0	0	0	0	0	0
A0161	10	8	6	16	6	6	6	6	6	6	6	6
A0162	2	1	1	1	1	1	1	1	1	1	1	1
A0171	0	0	0	0	0	0	0	0	0	0	0	0
A0176	6	2	2	2	2	2	2	2	2	2	2	2
A0060	4	4	0	0	0	0	0	0	0	0	0	0
A0154	16	12	8	8	8	8	8	8	8	8	8	8
A0237	6	6	2	2	2	2	2	2	2	2	2	2
A0090	4	4	4	4	3	2	2	2	2	3	2	2
A0618	16	0	0	0	0	0	0	0	0	0	0	0
A0934	19	19	19	19	19	19	38	38	0	0	0	0
C0818	3	3	3	7	5	3	3	3	3	3	3	3
C0822	4	3	3	7	4	3	3	3	3	3	3	3
C0823	5	5	2	2	2	2	2	2	2	2	2	2
C0848	2	2	2	4	2	2	2	2	2	2	2	2
C1314	0	0	0	0	0	0	0	0	0	0	0	0
C1362	2	2	2	3	2	2	2	2	2	2	2	2
C1364	6	6	22	16	10	6	6	6	22	22	22	22
C1366	3	3	3	5	3	3	3	3	3	3	3	3
C1369	5	5	5	13	13	5	5	5	5	5	5	5
C1382	5	5	5	4	4	2	3	3	2	2	2	2
C0206	33	24	4	4	4	4	4	4	4	4	4	4
C0995	1	1	0	0	0	0	0	0	0	0	0	0
C1486	1	1	0	0	0	0	0	0	0	0	0	0
C1488	1	1	0	0	0	0	0	0	0	0	0	0
C2211	2	2	2	6	6	2	2	2	2	2	2	2
C2226	13	7	5	5	5	5	5	5	5	5	5	5
C2422	1	1	0	0	0	0	0	0	0	0	0	0
C2425	0	0	0	1	0	0	0	0	0	0	0	0
C2503	2	1	0	0	0	0	0	0	0	0	0	0
C0255	27	27	7	7	7	33	33	33	59	59	59	59
C0259	11	11	11	87	11	11	11	11	32	32	32	32
C0061	3	3	0	0	0	0	0	0	0	0	0	0
C0002	2	2	2	4	3	2	2	2	2	3	2	2
C1243	3	3	3	5	4	3	3	3	3	4	3	3
C1945	1	1	1	1	1	1	0	0	0	0	0	0
C2759	0	0	0	0	0	0	0	0	0	0	0	0
C2845	1	1	1	1	1	1	24	24	0	0	0	0
C2871	0	0	0	8	0	0	0	0	0	0	0	0
C3079	8	2	2	2	2	2	2	2	2	2	2	2
C3289	0	0	0	2	0	0	0	0	0	0	0	0
C3462	40	40	40	40	0	0	0	0	0	0	0	0
C3613	1	0	0	0	0	0	0	0	0	0	0	0
C3640	0	0	0	30	0	0	0	0	0	0	0	0
C4119	0	0	0	0	0	0	0	0	0	0	0	0

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
C0072	1	1	0	0	0	0	0	0	0	0	0	0
C0273	6	6	0	6	0	0	0	0	0	0	0	0
C0018	1	1	1	2	1	1	1	1	1	1	1	1
C0075	24	20	16	20	20	16	16	16	16	20	16	16
C0269	0	0	0	0	0	0	0	0	0	0	0	0
C0281	30	30	30	30	30	30	342	342	342	342	342	342
C0035	1	1	0	0	0	0	0	0	0	0	0	0
C0074	1	1	0	0	0	0	0	0	0	0	0	0
C0225	0	0	0	0	0	0	0	0	0	0	0	0
G1203	1	1	1	1	1	1	1	1	1	1	1	1
G1218	0	0	0	0	0	0	4	0	0	0	0	0
G1295	8	8	8	16	12	8	8	8	8	12	8	8
G1297	7	7	7	15	11	7	7	7	7	11	7	7
G0286	0	0	0	0	0	0	0	0	0	0	0	0
G0472	1	1	1	1	1	1	1	1	1	1	1	1
G0473	2	2	2	2	2	2	2	2	2	2	2	2
G0475	0	0	0	0	0	0	0	0	0	0	0	0
G0476	0	0	0	0	0	0	0	0	0	0	0	0
G0595	0	0	0	0	0	0	0	0	0	0	0	0
G0596	0	0	0	0	0	0	0	0	0	0	0	0
G0597	0	0	0	0	0	0	0	0	0	0	0	0
G0598	0	0	0	0	0	0	0	0	0	0	0	0
G0599	0	0	0	0	0	0	0	0	0	0	0	0
G0732	0	0	0	0	0	0	0	0	0	0	0	0
G0733	0	0	0	0	0	0	0	0	0	0	0	0
G0734	0	0	0	0	0	0	0	0	0	0	0	0
G0835	4	4	2	2	2	2	2	2	2	2	2	2
G1095	2	2	2	2	2	2	2	2	2	2	2	2
G1878	1	1	1	2	1	1	1	1	1	1	1	1
G2002	8	8	8	8	8	8	8	8	20	20	20	20
G2004	8	8	8	8	8	8	8	8	20	20	20	20
G2343	0	0	0	0	0	0	0	0	0	0	0	0
G2415	1	1	1	1	1	1	1	1	1	1	1	1
G0288	5	5	5	12	5	5	13	8	5	7	5	5
G0023	2	2	2	2	2	2	2	2	2	2	2	2
G0062	1	1	1	1	0	0	0	0	0	0	0	0
G0109	2	2	2	4	3	2	2	2	2	3	2	2
G0110	2	2	2	3	3	2	2	2	2	3	2	2
G0181	0	0	0	0	0	0	0	0	0	0	0	0
G0212	1	1	1	1	1	1	1	1	1	1	1	1
G0213	1	1	1	1	1	1	1	1	1	1	1	1
G0164	784	32	32	32	32	32	152	152	152	152	152	152
G0171	20	20	20	20	20	75	75	75	75	75	75	75
G0765	2	2	2	2	2	2	2	2	2	2	2	2
G1043	27	20	13	32	29	26	26	26	26	29	26	26
G2197	85	85	13	13	134	134	134	134	134	134	134	134
G2427	1	1	1	1	1	1	1	1	1	1	1	1
G2428	1	1	1	1	1	1	1	1	1	1	1	1
G2517	0	0	0	0	0	0	0	0	0	0	0	0
G2978	1	1	1	1	1	1	1	1	1	1	1	1
G3063	4	4	4	4	4	4	4	4	4	4	4	4
G3131	6	6	6	6	6	6	6	6	6	6	6	6
G3135	1	1	1	1	1	1	1	1	1	1	1	1
G3142	0	0	0	0	0	0	4	4	0	0	0	0
G3775	2	2	2	2	2	2	2	2	2	2	2	2
G4248	1	1	1	1	1	1	1	1	1	1	1	1
G4391	9	9	9	9	9	9	9	9	9	9	9	9
G0082	2	2	2	2	2	2	2	2	2	2	2	2
G0089	2	2	2	2	2	2	2	2	2	2	2	2

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
G0100	6	6	6	13	6	6	14	9	6	14	12	12
G0073	1	1	1	1	1	1	0	0	0	0	0	0
G0094	0	0	0	0	0	0	2	2	2	2	2	2
G0097	0	0	0	0	0	0	0	0	0	0	0	0
G0099	0	0	0	0	0	0	0	0	0	0	0	0
G0111	0	0	0	0	0	0	0	0	0	0	0	0
G0096	1	1	1	1	1	1	1	1	1	1	1	1
G0097	1	1	1	1	1	1	1	1	1	1	1	1
G0089	12	12	5	12	5	5	5	5	5	5	5	5
G1376	11	11	5	32	5	5	5	5	5	5	5	5
G1762	31	27	8	8	8	8	29	29	29	29	29	29
G0373	6	6	9	9	9	12	12	12	15	15	15	18
Holding Cost \$	13039.72	12652.37	10497.3	17442.69	11188.02	9480.987	11557	11356.23	14477.66	16602.5	15151.41	15211.77

LAMPIRAN F

HASIL PENYELESAIAN STUDI KASUS

Berikut merupakan hasil *running* algoritma *Simulated Annealing* untuk memecahkan kasus *spare part inventory* pada PT INALUM (Persero).

Hasil *Re-Order Level* (c)

Item	<i>Re-order Level</i> (s)
A0116	0
A0391	9
A0408	5
A0417	5
A0419	0
A0510	0
A0531	5
A0533	10
A0650	2
A0668	4
A0812	0
A1025	0
A1216	5
A0022	0
A1857	0
A1860	2
A1876	2
A1910	4
A2276	5
A2300	4
A2319	6
A2322	3
A2330	4
A2753	0
A0185	0
A0704	0
A0705	0
A0742	0
A0753	3
A0099	1
A2230	7
A2600	1
A3336	3

Item	<i>Re-order Level</i> (s)
A4311	13
A0135	0
A0144	5
A0147	0
A0150	0
A0074	0
A0120	4
A0134	2
A0158	0
A0161	6
A0162	1
A0171	2
A0176	2
A0060	0
A0154	8
A0237	2
A0090	2
A0618	0
A0934	0
C0818	3
C0822	3
C0823	2
C0848	2
C1314	0
C1362	2
C1364	6
C1366	3
C1369	5
C1382	2
C0206	4
C0995	2
C1486	2
C1488	2

Item	<i>Re-order Level</i> (s)
C2211	2
C2226	5
C2422	0
C2425	0
C2503	0
C0255	7
C0259	11
C0061	0
C0002	2
C1243	3
C1945	0
C2759	0
C2845	0
C2871	0
C3079	2
C3289	0
C3462	0
C3613	0
C3640	0
C4119	0
C0072	0
C0273	0
C0018	1
C0075	16
C0269	0
C0281	30
C0035	0
C0074	0
C0225	2
G1203	1
G1218	0
G1295	8
G1297	7

Item	<i>Re-order Level (s)</i>
G0286	2
G0472	1
G0473	2
G0475	0
G0476	0
G0595	2
G0596	3
G0597	3
G0598	2
G0599	4
G0732	2
G0733	2
G0734	5
G0835	2
G1095	2
G1878	1
G2002	8
G2004	8
G2343	2
G2415	1

Item	<i>Re-order Level (s)</i>
G0288	5
G0023	2
G0062	5
G0109	2
G0110	2
G0181	9
G0212	1
G0213	1
G0164	32
G0171	0
G0765	2
G1043	13
G2197	13
G2427	1
G2428	1
G2517	3
G2978	1
G3063	4
G3131	6
G3135	1

Item	<i>Re-order Level (s)</i>
G3142	8
G3775	2
G4248	1
G4391	9
G0082	2
G0089	2
G0100	6
G0073	0
G0094	0
G0097	0
G0099	0
G0111	0
G0096	1
G0097	1
G0089	5
G1376	5
G1762	8
G0373	0

Hasil Can-Order Level (c)

Can-order Level (c)	A0 116	A0 391	A0 408	A0 417	A0 419	A0 510	A0 531	A0 533	A0 650	A0 668	A0 812	A1 025	A1 216	A0 022	A1 857	A1 860	A1 876	A1 910	A2 276	A2 300	A2 319	A2 322	A2 330	A2 753	A0 185	A0 704
A0116	0	11	5	6	0	0	6	12	2	4	0	0	6	0	0	2	2	5	6	5	7	4	4	0	0	0
A0391	0	9	6	5	0	0	6	10	2	4	0	0	5	0	0	2	2	5	6	5	6	4	5	0	0	0
A0408	0	11	5	6	0	0	6	11	2	5	0	0	5	0	0	2	2	5	6	5	7	3	4	0	0	0
A0417	0	9	5	5	0	0	5	10	2	4	0	0	6	0	0	2	2	4	5	4	6	4	4	0	0	0
A0419	0	9	5	5	0	0	6	12	2	5	0	0	5	0	0	2	2	4	5	5	7	4	5	0	0	0
A0510	0	11	5	6	0	0	6	11	2	5	0	0	6	0	0	2	2	4	6	5	7	3	4	0	0	0
A0531	0	11	5	5	0	0	5	11	2	4	0	0	6	0	0	2	2	4	6	5	7	4	4	0	0	0
A0533	0	10	5	6	0	0	5	10	2	4	0	0	6	0	0	2	2	4	6	4	6	3	5	0	0	0
A0650	0	10	5	6	0	0	6	11	2	4	0	0	6	0	0	2	2	4	6	5	7	3	4	0	0	0
A0668	0	9	6	6	0	0	5	11	2	4	0	0	5	0	0	2	2	5	6	5	6	4	4	0	0	0
A0812	0	11	6	5	0	0	5	12	2	5	0	0	5	0	0	2	2	5	5	4	7	4	4	0	0	0
A1025	0	9	5	5	0	0	5	11	2	5	0	0	5	0	0	2	2	5	5	5	6	3	5	0	0	0
A1216	0	11	6	5	0	0	6	11	2	5	0	0	5	0	0	2	2	5	5	4	7	3	4	0	0	0
A0022	0	9	5	5	0	0	6	10	2	5	0	0	6	0	0	2	2	4	5	5	6	3	5	0	0	0
A1857	0	10	5	6	0	0	5	10	2	5	0	0	5	0	0	2	2	5	5	5	7	4	4	0	0	0
A1860	0	11	6	5	0	0	6	12	2	5	0	0	5	0	0	2	2	4	6	5	7	4	4	0	0	0
A1876	0	10	6	6	0	0	5	11	2	5	0	0	6	0	0	2	2	4	5	5	7	4	5	0	0	0
A1910	0	11	6	5	0	0	6	11	2	4	0	0	5	0	0	2	2	4	6	5	7	4	4	0	0	0
A2276	0	10	5	6	0	0	6	12	2	5	0	0	6	0	0	2	2	4	5	5	6	4	5	0	0	0
A2300	0	10	5	6	0	0	6	10	2	5	0	0	5	0	0	2	2	5	5	4	7	4	5	0	0	0
A2319	0	9	5	6	0	0	6	12	2	5	0	0	6	0	0	2	2	5	6	4	6	3	5	0	0	0
A2322	0	10	5	5	0	0	5	10	2	5	0	0	5	0	0	2	2	5	5	4	7	3	4	0	0	0
A2330	0	11	5	6	0	0	6	10	2	4	0	0	5	0	0	2	2	4	5	4	7	4	4	0	0	0
A2753	0	10	5	5	0	0	6	10	2	4	0	0	6	0	0	2	2	5	5	4	6	4	5	0	0	0
A0185	0	9	5	6	0	0	6	11	2	5	0	0	6	0	0	2	2	5	6	4	6	4	5	0	0	0
A0704	0	11	6	5	0	0	5	11	2	5	0	0	6	0	0	2	2	5	6	5	6	4	4	0	0	0
A0705	0	10	6	6	0	0	5	11	2	4	0	0	5	0	0	2	2	5	6	5	6	4	5	0	0	0
A0742	0	10	6	6	0	0	5	10	2	4	0	0	6	0	0	2	2	5	5	4	7	3	4	0	0	0
A0753	0	11	5	5	0	0	6	10	2	5	0	0	5	0	0	2	2	4	5	5	6	4	4	0	0	0
A0099	0	11	5	6	0	0	6	11	2	5	0	0	6	0	0	2	2	5	5	4	7	3	4	0	0	0
A2230	0	9	5	5	0	0	6	12	2	5	0	0	5	0	0	2	2	4	5	4	7	4	5	0	0	0
A2600	0	9	5	5	0	0	5	11	2	4	0	0	5	0	0	2	2	4	6	5	7	4	5	0	0	0
A3336	0	10	5	6	0	0	5	12	2	4	0	0	5	0	0	2	2	4	6	4	7	4	4	0	0	0
A4311	0	10	6	6	0	0	6	10	2	5	0	0	5	0	0	2	2	5	5	4	7	4	4	0	0	0
A0135	0	9	6	6	0	0	5	11	2	4	0	0	5	0	0	2	2	4	5	4	6	3	4	0	0	0
A0144	0	10	5	6	0	0	5	12	2	4	0	0	6	0	0	2	2	5	5	5	6	3	4	0	0	0
A0147	0	9	6	6	0	0	6	12	2	5	0	0	6	0	0	2	2	5	5	4	7	4	5	0	0	0
A0150	0	11	6	6	0	0	6	11	2	4	0	0	6	0	0	2	2	5	6	4	7	4	5	0	0	0
A0074	0	9	6	6	0	0	5	10	2	5	0	0	5	0	0	2	2	5	6	4	7	3	4	0	0	0

Can-order Level (c)	A0 116	A0 391	A0 408	A0 417	A0 419	A0 510	A0 531	A0 533	A0 650	A0 668	A0 812	A1 025	A1 216	A0 022	A1 857	A1 860	A1 876	A1 910	A2 276	A2 300	A2 319	A2 322	A2 330	A2 753	A0 185	A0 704
A0120	0	9	6	6	0	0	6	10	2	4	0	0	6	0	0	2	2	5	5	5	6	4	4	0	0	0
A0134	0	11	5	5	0	0	5	12	2	5	0	0	5	0	0	2	2	4	6	4	7	3	4	0	0	0
A0158	0	11	5	6	0	0	5	12	2	5	0	0	5	0	0	2	2	5	6	4	6	3	4	0	0	0
A0161	0	11	5	5	0	0	5	10	2	5	0	0	6	0	0	2	2	4	5	5	6	4	5	0	0	0
A0162	0	11	5	6	0	0	6	11	2	5	0	0	5	0	0	2	2	5	6	4	6	4	4	0	0	0
A0171	0	9	5	6	0	0	6	10	2	4	0	0	5	0	0	2	2	5	6	5	7	3	4	0	0	0
A0176	0	10	6	5	0	0	6	11	2	4	0	0	5	0	0	2	2	5	5	4	6	3	4	0	0	0
A0060	0	10	5	6	0	0	6	10	2	5	0	0	6	0	0	2	2	5	6	5	6	3	5	0	0	0
A0154	0	11	6	5	0	0	5	12	2	5	0	0	5	0	0	2	2	5	5	4	7	3	4	0	0	0
A0237	0	11	6	6	0	0	5	12	2	4	0	0	5	0	0	2	2	5	5	4	7	3	5	0	0	0
A0090	0	9	6	6	0	0	5	11	2	5	0	0	5	0	0	2	2	4	6	4	6	3	5	0	0	0
A0618	0	11	6	6	0	0	5	10	2	5	0	0	5	0	0	2	2	4	6	5	6	3	4	0	0	0
A0934	0	10	5	5	0	0	5	12	2	4	0	0	6	0	0	2	2	4	6	5	6	3	4	0	0	0

Can-order Level (c)	A0 705	A0 742	A0 753	A0 099	A2 230	A2 600	A3 336	A4 311	A0 135	A0 144	A0 147	A0 150	A0 074	A0 120	A0 134	A0 158	A0 161	A0 162	A0 171	A0 176	A0 060	A0 154	A0 237	A0 090	A0 618	A0 934
A0116	0	0	4	1	8	1	4	14	0	6	0	0	0	4	2	0	7	1	2	2	0	8	2	2	0	0
A0391	0	0	3	1	7	1	3	15	0	5	0	0	0	4	2	0	7	1	2	2	0	8	2	2	0	0
A0408	0	0	3	1	7	1	3	14	0	5	0	0	0	5	2	0	6	1	2	2	0	9	2	2	0	0
A0417	0	0	4	1	8	1	3	14	0	5	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A0419	0	0	3	1	7	1	4	15	0	5	0	0	0	5	2	0	6	1	2	2	0	8	2	2	0	0
A0510	0	0	4	1	8	1	3	16	0	5	0	0	0	4	2	0	7	1	2	2	0	8	2	2	0	0
A0531	0	0	3	1	8	1	3	14	0	5	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A0533	0	0	4	1	7	1	3	14	0	5	0	0	0	4	2	0	7	1	2	2	0	9	2	2	0	0
A0650	0	0	3	1	7	1	3	14	0	6	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A0668	0	0	3	1	7	1	3	15	0	5	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A0812	0	0	3	1	7	1	3	14	0	6	0	0	0	5	2	0	7	1	2	2	0	9	2	2	0	0
A1025	0	0	4	1	8	1	3	16	0	5	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A1216	0	0	3	1	8	1	4	15	0	6	0	0	0	4	2	0	6	1	2	2	0	10	2	2	0	0
A0022	0	0	3	1	8	1	4	16	0	6	0	0	0	4	2	0	7	1	2	2	0	9	2	2	0	0
A1857	0	0	3	1	7	1	3	15	0	5	0	0	0	5	2	0	7	1	2	2	0	9	2	2	0	0
A1860	0	0	4	1	7	1	4	14	0	5	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A1876	0	0	4	1	7	1	3	13	0	5	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A1910	0	0	3	1	8	1	3	15	0	5	0	0	0	4	2	0	6	1	2	2	0	8	2	2	0	0
A2276	0	0	3	1	8	1	4	16	0	6	0	0	0	4	2	0	6	1	2	2	0	10	2	2	0	0
A2300	0	0	4	1	8	1	4	15	0	6	0	0	0	4	2	0	6	1	2	2	0	8	2	2	0	0
A2319	0	0	4	1	7	1	3	16	0	5	0	0	0	5	2	0	6	1	2	2	0	10	2	2	0	0
A2322	0	0	4	1	7	1	4	14	0	5	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A2330	0	0	4	1	8	1	3	15	0	6	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A2753	0	0	3	1	8	1	3	16	0	6	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0

<i>Can-order Level (c)</i>	A0 705	A0 742	A0 753	A0 099	A2 230	A2 600	A3 336	A4 311	A0 135	A0 144	A0 147	A0 150	A0 074	A0 120	A0 134	A0 158	A0 161	A0 162	A0 171	A0 176	A0 060	A0 154	A0 237	A0 090	A0 618	A0 934
A0185	0	0	4	1	7	1	4	15	0	6	0	0	0	5	2	0	6	1	2	2	0	10	2	2	0	0
A0704	0	0	4	1	7	1	4	16	0	5	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A0705	0	0	4	1	7	1	3	14	0	6	0	0	0	4	2	0	7	1	2	2	0	9	2	2	0	0
A0742	0	0	3	1	8	1	4	14	0	6	0	0	0	4	2	0	6	1	2	2	0	10	2	2	0	0
A0753	0	0	3	1	7	1	4	14	0	5	0	0	0	4	2	0	6	1	2	2	0	10	2	2	0	0
A0099	0	0	4	1	7	1	3	13	0	5	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A2230	0	0	4	1	7	1	3	16	0	6	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A2600	0	0	4	1	8	1	3	14	0	5	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A3336	0	0	3	1	8	1	3	14	0	5	0	0	0	4	2	0	7	1	2	2	0	10	2	2	0	0
A4311	0	0	3	1	8	1	4	13	0	5	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A0135	0	0	4	1	8	1	3	16	0	6	0	0	0	5	2	0	6	1	2	2	0	9	2	2	0	0
A0144	0	0	3	1	7	1	4	13	0	5	0	0	0	4	2	0	7	1	2	2	0	10	2	2	0	0
A0147	0	0	4	1	7	1	4	13	0	6	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A0150	0	0	3	1	7	1	4	15	0	5	0	0	0	5	2	0	7	1	2	2	0	9	2	2	0	0
A0074	0	0	3	1	7	1	3	16	0	6	0	0	0	4	2	0	6	1	2	2	0	8	2	2	0	0
A0120	0	0	4	1	8	1	3	14	0	5	0	0	0	4	2	0	7	1	2	2	0	10	2	2	0	0
A0134	0	0	4	1	7	1	3	13	0	6	0	0	0	5	2	0	7	1	2	2	0	10	2	2	0	0
A0158	0	0	4	1	7	1	3	13	0	6	0	0	0	5	2	0	6	1	2	2	0	9	2	2	0	0
A0161	0	0	3	1	8	1	3	15	0	6	0	0	0	4	2	0	6	1	2	2	0	9	2	2	0	0
A0162	0	0	4	1	8	1	3	15	0	5	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A0171	0	0	4	1	8	1	3	13	0	6	0	0	0	5	2	0	6	1	2	2	0	8	2	2	0	0
A0176	0	0	3	1	7	1	3	15	0	6	0	0	0	4	2	0	7	1	2	2	0	10	2	2	0	0
A0060	0	0	3	1	7	1	3	13	0	5	0	0	0	4	2	0	6	1	2	2	0	10	2	2	0	0
A0154	0	0	4	1	7	1	4	15	0	6	0	0	0	5	2	0	7	1	2	2	0	8	2	2	0	0
A0237	0	0	4	1	7	1	3	13	0	6	0	0	0	5	2	0	6	1	2	2	0	10	2	2	0	0
A0090	0	0	4	1	7	1	3	16	0	5	0	0	0	4	2	0	7	1	2	2	0	10	2	2	0	0
A0618	0	0	4	1	8	1	3	15	0	6	0	0	0	4	2	0	7	1	2	2	0	9	2	2	0	0
A0934	0	0	4	1	7	1	4	16	0	5	0	0	0	5	2	0	6	1	2	2	0	8	2	2	0	0

<i>Can-order Level (c)</i>	C081 8	C082 2	C082 3	C084 8	C131 4	C136 2	C136 4	C136 6	C136 9	C138 2	C020 6	C099 5	C148 6	C148 8	C221 1	C222 6	C242 2	C242 5	C250 3	C025 5	C025 9
C0818	3	3	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	8	13
C0822	4	3	2	2	0	2	6	4	5	2	5	2	2	2	2	6	0	0	0	8	11
C0823	3	4	2	2	0	2	6	4	5	2	5	2	2	2	2	5	0	0	0	7	12
C0848	3	3	2	2	0	2	7	3	5	2	4	2	2	2	2	6	0	0	0	8	12
C1314	4	4	2	2	0	2	6	4	6	2	4	2	2	2	2	5	0	0	0	8	12
C1362	4	3	2	2	0	2	7	4	6	2	5	2	2	2	2	6	0	0	0	7	13
C1364	4	3	2	2	0	2	6	3	5	2	4	2	2	2	2	6	0	0	0	7	13
C1366	3	3	2	2	0	2	6	3	6	2	4	2	2	2	2	5	0	0	0	7	13

<i>Can-order Level</i> (c)	C081 8	C082 2	C082 3	C084 8	C131 4	C136 2	C136 4	C136 6	C136 9	C138 2	C020 6	C099 5	C148 6	C148 8	C221 1	C222 6	C242 2	C242 5	C250 3	C025 5	C025 9
C1369	3	4	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	8	13
C1382	4	4	2	2	0	2	7	4	6	2	4	2	2	2	2	5	0	0	0	7	13
C0206	4	4	2	2	0	2	6	4	6	2	4	2	2	2	2	6	0	0	0	7	11
C0995	3	4	2	2	0	2	6	4	6	2	4	2	2	2	2	5	0	0	0	7	12
C1486	3	4	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	8	11
C1488	4	3	2	2	0	2	7	3	6	2	5	2	2	2	2	6	0	0	0	8	12
C2211	3	3	2	2	0	2	6	3	5	2	5	2	2	2	2	6	0	0	0	8	13
C2226	3	4	2	2	0	2	6	4	5	2	5	2	2	2	2	5	0	0	0	7	11
C2422	4	3	2	2	0	2	7	3	6	2	4	2	2	2	2	6	0	0	0	7	12
C2425	3	3	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	8	11
C2503	4	3	2	2	0	2	7	4	5	2	4	2	2	2	2	6	0	0	0	7	13
C0255	4	3	2	2	0	2	7	3	6	2	5	2	2	2	2	6	0	0	0	7	13
C0259	4	4	2	2	0	2	6	3	5	2	5	2	2	2	2	5	0	0	0	7	11
C0061	4	3	2	2	0	2	7	4	6	2	5	2	2	2	2	5	0	0	0	7	12
C0002	4	4	2	2	0	2	7	4	6	2	4	2	2	2	2	6	0	0	0	8	11
C1243	4	4	2	2	0	2	6	4	6	2	5	2	2	2	2	5	0	0	0	7	13
C1945	4	3	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	7	12
C2759	3	4	2	2	0	2	6	3	5	2	4	2	2	2	2	6	0	0	0	7	12
C2845	3	3	2	2	0	2	6	4	5	2	4	2	2	2	2	6	0	0	0	7	11
C2871	3	3	2	2	0	2	7	3	6	2	5	2	2	2	2	6	0	0	0	7	13
C3079	4	4	2	2	0	2	6	3	6	2	5	2	2	2	2	5	0	0	0	7	12
C3289	4	3	2	2	0	2	7	4	5	2	5	2	2	2	2	5	0	0	0	7	12
C3462	3	4	2	2	0	2	6	3	6	2	4	2	2	2	2	5	0	0	0	8	11
C3613	4	4	2	2	0	2	7	3	6	2	5	2	2	2	2	5	0	0	0	8	13
C3640	4	4	2	2	0	2	6	4	6	2	4	2	2	2	2	5	0	0	0	7	13
C4119	4	3	2	2	0	2	7	4	5	2	4	2	2	2	2	6	0	0	0	7	13
C0072	3	3	2	2	0	2	7	3	5	2	4	2	2	2	2	6	0	0	0	8	11
C0273	4	3	2	2	0	2	6	4	6	2	4	2	2	2	2	5	0	0	0	8	12
C0018	4	3	2	2	0	2	7	3	5	2	5	2	2	2	2	6	0	0	0	8	11
C0075	4	4	2	2	0	2	7	4	5	2	5	2	2	2	2	5	0	0	0	8	13
C0269	4	3	2	2	0	2	7	4	5	2	4	2	2	2	2	6	0	0	0	8	11
C0281	3	3	2	2	0	2	7	4	5	2	4	2	2	2	2	6	0	0	0	8	13
C0035	4	4	2	2	0	2	6	4	6	2	5	2	2	2	2	5	0	0	0	8	12
C0074	3	3	2	2	0	2	7	3	5	2	4	2	2	2	2	5	0	0	0	7	11
C0225	4	4	2	2	0	2	6	4	5	2	5	2	2	2	2	5	0	0	0	7	11

<i>Can-order Level (c)</i>	C00 61	C00 02	C12 43	C19 45	C27 59	C28 45	C28 71	C30 79	C32 89	C34 62	C36 13	C36 40	C41 19	C00 72	C02 73	C00 18	C00 75	C02 69	C02 81	C00 35	C00 74	C02 25
C0818	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	30	0	0	2
C0822	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	17	0	30	0	0	2
C0823	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	36	0	0	2
C0848	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	34	0	0	2
C1314	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	32	0	0	2
C1362	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	35	0	0	2
C1364	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	33	0	0	2
C1366	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	33	0	0	2
C1369	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	34	0	0	2
C1382	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	31	0	0	2
C0206	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	36	0	0	2
C0995	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	36	0	0	2
C1486	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	17	0	31	0	0	2
C1488	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	17	0	36	0	0	2
C2211	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	30	0	0	2
C2226	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	17	0	31	0	0	2
C2422	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	32	0	0	2
C2425	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	32	0	0	2
C2503	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	33	0	0	2
C0255	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	31	0	0	2
C0259	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	34	0	0	2
C0061	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	17	0	30	0	0	2
C0002	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	36	0	0	2
C1243	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	31	0	0	2
C1945	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	36	0	0	2
C2759	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	33	0	0	2
C2845	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	30	0	0	2
C2871	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	34	0	0	2
C3079	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	32	0	0	2
C3289	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	31	0	0	2
C3462	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	34	0	0	2
C3613	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	36	0	0	2
C3640	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	34	0	0	2
C4119	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	32	0	0	2
C0072	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	35	0	0	2
C0273	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	19	0	31	0	0	2
C0018	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	35	0	0	2
C0075	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	31	0	0	2
C0269	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	34	0	0	2
C0281	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	30	0	0	2
C0035	0	2	3	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	36	0	0	2

Can-order Level (c)	C00 61	C00 02	C12 43	C19 45	C27 59	C28 45	C28 71	C30 79	C32 89	C34 62	C36 13	C36 40	C41 19	C00 72	C02 73	C00 18	C00 75	C02 69	C02 81	C00 35	C00 74	C02 25
C0074	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	18	0	33	0	0	2
C0225	0	2	4	0	0	0	0	2	0	0	0	0	0	0	0	1	16	0	32	0	0	2

Can-order Level (c)	G 12 03	G 12 18	G 12 95	G 12 97	G 02 86	G 04 72	G 04 73	G 04 75	G 04 76	G 05 95	G 05 96	G 05 97	G 05 98	G 05 99	G 07 32	G 07 33	G 07 34	G 08 35	G 10 95	G 18 78	G 20 02	G 20 04	G 23 43	G 24 15	G 02 88	G 00 23	G 00 62	G 01 09	G 01 10	G 01 81	G 02 12
G1203	1	0	9	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	9	9	2	1	5	2	5	2	2	10	1
G1218	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	8	10	2	1	5	2	6	2	2	10	1
G1295	1	0	8	7	2	1	2	0	0	2	4	4	2	4	2	2	5	2	2	1	8	9	2	1	5	2	6	2	2	9	1
G1297	1	0	10	7	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	9	2	1	6	2	5	2	2	10	1
G0286	1	0	10	7	2	1	2	0	0	2	3	4	2	5	2	2	6	2	2	1	10	9	2	1	5	2	5	2	2	11	1
G0472	1	0	10	8	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	9	8	2	1	5	2	6	2	2	10	1
G0473	1	0	9	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	8	9	2	1	5	2	6	2	2	11	1
G0475	1	0	10	7	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	8	9	2	1	5	2	5	2	2	9	1
G0476	1	0	10	7	2	1	2	0	0	2	4	4	2	4	2	2	5	2	2	1	8	10	2	1	5	2	6	2	2	10	1
G0595	1	0	10	8	2	1	2	0	0	2	3	4	2	5	2	2	5	2	2	1	9	9	2	1	5	2	5	2	2	11	1
G0596	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	10	9	2	1	5	2	6	2	2	11	1
G0597	1	0	9	8	2	1	2	0	0	2	4	3	2	4	2	2	5	2	2	1	10	10	2	1	5	2	5	2	2	9	1
G0598	1	0	10	7	2	1	2	0	0	2	3	4	2	5	2	2	6	2	2	1	8	9	2	1	5	2	5	2	2	10	1
G0599	1	0	9	8	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	8	8	2	1	5	2	5	2	2	11	1
G0732	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	9	8	2	1	5	2	6	2	2	9	1
G0733	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	8	10	2	1	6	2	5	2	2	11	1
G0734	1	0	8	7	2	1	2	0	0	2	4	3	2	4	2	2	5	2	2	1	10	10	2	1	6	2	6	2	2	10	1
G0835	1	0	10	7	2	1	2	0	0	2	3	4	2	5	2	2	5	2	2	1	10	9	2	1	5	2	6	2	2	9	1
G1095	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	8	8	2	1	5	2	5	2	2	9	1
G1878	1	0	9	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	8	10	2	1	6	2	5	2	2	11	1
G2002	1	0	9	7	2	1	2	0	0	2	3	3	2	5	2	2	5	2	2	1	8	8	2	1	5	2	5	2	2	10	1
G2004	1	0	9	7	2	1	2	0	0	2	4	4	2	4	2	2	6	2	2	1	9	8	2	1	6	2	5	2	2	10	1
G2343	1	0	8	7	2	1	2	0	0	2	3	4	2	5	2	2	6	2	2	1	8	8	2	1	6	2	5	2	2	10	1
G2415	1	0	8	7	2	1	2	0	0	2	3	4	2	5	2	2	5	2	2	1	8	8	2	1	5	2	6	2	2	11	1
G0288	1	0	8	7	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	10	8	2	1	5	2	6	2	2	9	1
G0023	1	0	9	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	10	2	1	6	2	6	2	2	10	1
G0062	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	9	8	2	1	5	2	5	2	2	10	1
G0109	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	8	2	1	6	2	5	2	2	10	1
G0110	1	0	10	8	2	1	2	0	0	2	3	4	2	5	2	2	5	2	2	1	10	10	2	1	5	2	6	2	2	11	1
G0181	1	0	8	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	10	9	2	1	6	2	5	2	2	9	1
G0212	1	0	10	8	2	1	2	0	0	2	4	4	2	5	2	2	5	2	2	1	10	9	2	1	5	2	6	2	2	11	1
G0213	1	0	10	7	2	1	2	0	0	2	3	3	2	5	2	2	5	2	2	1	9	9	2	1	6	2	5	2	2	10	1
G0164	1	0	8	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	8	10	2	1	5	2	5	2	2	11	1

Can- order Level (c)	G 12 03	G 12 18	G 12 95	G 12 97	G 02 86	G 04 72	G 04 73	G 04 75	G 04 76	G 05 95	G 05 96	G 05 97	G 05 98	G 05 99	G 07 32	G 07 33	G 07 34	G 08 35	G 10 95	G 18 78	G 20 02	G 20 04	G 23 43	G 24 15	G 02 88	G 00 23	G 00 62	G 01 09	G 01 10	G 01 81	G 02 12
G0171	1	0	9	7	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	9	9	2	1	5	2	5	2	2	10	1
G0765	1	0	8	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	8	10	2	1	6	2	5	2	2	10	1
G1043	1	0	9	7	2	1	2	0	0	2	3	4	2	5	2	2	5	2	2	1	9	10	2	1	5	2	5	2	2	9	1
G2197	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	9	2	1	5	2	5	2	2	11	1
G2427	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	8	8	2	1	5	2	5	2	2	9	1
G2428	1	0	9	7	2	1	2	0	0	2	3	4	2	5	2	2	6	2	2	1	9	9	2	1	6	2	6	2	2	11	1
G2517	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	10	10	2	1	6	2	6	2	2	11	1
G2978	1	0	9	8	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	9	9	2	1	6	2	5	2	2	10	1
G3063	1	0	9	7	2	1	2	0	0	2	4	3	2	5	2	2	6	2	2	1	8	9	2	1	6	2	6	2	2	10	1
G3131	1	0	8	7	2	1	2	0	0	2	3	3	2	5	2	2	6	2	2	1	10	8	2	1	6	2	6	2	2	11	1
G3135	1	0	8	7	2	1	2	0	0	2	4	3	2	4	2	2	5	2	2	1	10	8	2	1	5	2	5	2	2	11	1
G3142	1	0	10	7	2	1	2	0	0	2	4	4	2	5	2	2	5	2	2	1	10	8	2	1	6	2	5	2	2	11	1
G3775	1	0	8	8	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	9	8	2	1	6	2	6	2	2	10	1
G4248	1	0	9	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	9	2	1	5	2	6	2	2	10	1
G4391	1	0	8	8	2	1	2	0	0	2	4	3	2	4	2	2	6	2	2	1	8	9	2	1	6	2	6	2	2	9	1
G0082	1	0	8	7	2	1	2	0	0	2	4	4	2	4	2	2	5	2	2	1	8	9	2	1	5	2	6	2	2	10	1
G0089	1	0	8	7	2	1	2	0	0	2	4	4	2	4	2	2	6	2	2	1	8	10	2	1	6	2	5	2	2	10	1
G0100	1	0	9	7	2	1	2	0	0	2	4	3	2	4	2	2	5	2	2	1	10	8	2	1	5	2	6	2	2	10	1
G0073	1	0	10	8	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	9	10	2	1	6	2	5	2	2	10	1
G0094	1	0	9	7	2	1	2	0	0	2	4	4	2	5	2	2	5	2	2	1	10	10	2	1	5	2	6	2	2	9	1
G0097	1	0	8	8	2	1	2	0	0	2	4	3	2	5	2	2	5	2	2	1	10	10	2	1	6	2	5	2	2	10	1
G0099	1	0	8	8	2	1	2	0	0	2	4	3	2	5	2	2	5	2	2	1	8	9	2	1	5	2	5	2	2	10	1
G0111	1	0	8	7	2	1	2	0	0	2	4	3	2	5	2	2	5	2	2	1	8	10	2	1	5	2	6	2	2	11	1
G0096	1	0	10	8	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	9	9	2	1	5	2	5	2	2	9	1
G0097	1	0	9	8	2	1	2	0	0	2	4	3	2	4	2	2	5	2	2	1	10	9	2	1	6	2	5	2	2	9	1
G0089	1	0	8	7	2	1	2	0	0	2	3	4	2	4	2	2	5	2	2	1	10	9	2	1	5	2	6	2	2	11	1
G1376	1	0	9	8	2	1	2	0	0	2	4	4	2	5	2	2	5	2	2	1	10	8	2	1	5	2	5	2	2	10	1
G1762	1	0	8	7	2	1	2	0	0	2	3	3	2	4	2	2	5	2	2	1	8	9	2	1	5	2	6	2	2	11	1
G0373	1	0	8	8	2	1	2	0	0	2	3	4	2	4	2	2	6	2	2	1	9	10	2	1	6	2	6	2	2	10	1

Can- order Level (c)	G 02 13	G 01 64	G 01 71	G 07 65	G 10 43	G 21 97	G 24 27	G 24 28	G 25 17	G 29 78	G 30 63	G 31 31	G 31 35	G 31 42	G 37 75	G 42 48	G 43 91	G 00 82	G 00 89	G 01 00	G 00 73	G 00 94	G 00 97	G 00 99	G 01 11	G 00 96	G 00 97	G 00 89	G 13 76	G 17 62	G 03 73
G1203	1	32	0	2	15	13	1	1	4	1	4	7	1	9	2	1	10	2	2	6	0	0	0	0	0	1	1	5	6	10	0
G1218	1	33	0	2	16	16	1	1	4	1	5	7	1	9	2	1	11	2	2	7	0	0	0	0	0	1	1	6	6	9	0
G1295	1	32	0	2	16	13	1	1	4	1	4	6	1	8	2	1	10	2	2	6	0	0	0	0	0	1	1	5	5	9	0
G1297	1	34	0	2	14	16	1	1	4	1	5	7	1	10	2	1	10	2	2	7	0	0	0	0	0	1	1	6	6	10	0

Can- order Level (c)	G 02 13	G 01 64	G 01 71	G 07 65	G 10 43	G 21 97	G 24 27	G 24 28	G 25 17	G 29 78	G 30 63	G 31 31	G 31 35	G 31 42	G 37 75	G 42 48	G 43 91	G 00 82	G 00 89	G 01 00	G 00 73	G 00 94	G 00 97	G 00 99	G 01 11	G 00 96	G 00 97	G 00 89	G 13 76	G 17 62	G 03 73
G0286	1	34	0	2	13	13	1	1	3	1	5	7	1	8	2	1	10	2	2	6	0	0	0	0	0	1	1	5	5	8	0
G0472	1	32	0	2	16	14	1	1	3	1	4	6	1	9	2	1	9	2	2	6	0	0	0	0	0	1	1	5	5	9	0
G0473	1	32	0	2	14	16	1	1	4	1	4	7	1	8	2	1	9	2	2	6	0	0	0	0	0	1	1	6	6	8	0
G0475	1	34	0	2	14	14	1	1	4	1	4	6	1	8	2	1	11	2	2	7	0	0	0	0	0	1	1	6	5	9	0
G0476	1	37	0	2	16	15	1	1	3	1	4	7	1	9	2	1	9	2	2	7	0	0	0	0	0	1	1	5	5	8	0
G0595	1	37	0	2	15	15	1	1	3	1	5	6	1	8	2	1	10	2	2	7	0	0	0	0	0	1	1	5	6	8	0
G0596	1	33	0	2	14	13	1	1	3	1	4	7	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	5	5	10	0
G0597	1	34	0	2	16	15	1	1	4	1	5	7	1	10	2	1	11	2	2	7	0	0	0	0	0	1	1	5	5	10	0
G0598	1	38	0	2	15	16	1	1	4	1	4	6	1	9	2	1	11	2	2	7	0	0	0	0	0	1	1	6	6	10	0
G0599	1	32	0	2	16	13	1	1	4	1	5	7	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	5	5	10	0
G0732	1	35	0	2	14	16	1	1	4	1	5	7	1	8	2	1	9	2	2	7	0	0	0	0	0	1	1	5	6	8	0
G0733	1	32	0	2	13	13	1	1	4	1	5	6	1	10	2	1	10	2	2	7	0	0	0	0	0	1	1	5	6	8	0
G0734	1	35	0	2	16	14	1	1	3	1	5	6	1	9	2	1	10	2	2	7	0	0	0	0	0	1	1	6	6	8	0
G0835	1	32	0	2	16	13	1	1	4	1	5	7	1	10	2	1	9	2	2	6	0	0	0	0	0	1	1	5	5	9	0
G1095	1	35	0	2	13	16	1	1	4	1	5	6	1	9	2	1	9	2	2	7	0	0	0	0	0	1	1	5	6	8	0
G1878	1	37	0	2	15	14	1	1	4	1	4	7	1	9	2	1	10	2	2	7	0	0	0	0	0	1	1	5	5	8	0
G2002	1	33	0	2	15	13	1	1	3	1	5	7	1	8	2	1	9	2	2	6	0	0	0	0	0	1	1	5	5	9	0
G2004	1	35	0	2	16	16	1	1	4	1	4	7	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	5	6	8	0
G2343	1	38	0	2	14	16	1	1	3	1	4	6	1	9	2	1	9	2	2	6	0	0	0	0	0	1	1	5	5	8	0
G2415	1	34	0	2	15	15	1	1	3	1	5	6	1	8	2	1	10	2	2	6	0	0	0	0	0	1	1	6	5	8	0
G0288	1	38	0	2	14	16	1	1	3	1	4	7	1	9	2	1	9	2	2	6	0	0	0	0	0	1	1	5	6	10	0
G0023	1	37	0	2	14	14	1	1	4	1	4	6	1	10	2	1	10	2	2	7	0	0	0	0	0	1	1	6	6	8	0
G0062	1	36	0	2	15	15	1	1	4	1	5	6	1	9	2	1	11	2	2	7	0	0	0	0	0	1	1	5	6	9	0
G0109	1	35	0	2	15	15	1	1	3	1	5	7	1	9	2	1	10	2	2	6	0	0	0	0	0	1	1	5	6	9	0
G0110	1	38	0	2	13	15	1	1	4	1	5	6	1	9	2	1	10	2	2	6	0	0	0	0	0	1	1	6	6	10	0
G0181	1	34	0	2	14	14	1	1	4	1	5	6	1	8	2	1	11	2	2	6	0	0	0	0	0	1	1	5	5	10	0
G0212	1	35	0	2	16	16	1	1	3	1	4	7	1	8	2	1	10	2	2	6	0	0	0	0	0	1	1	5	6	8	0
G0213	1	38	0	2	14	15	1	1	3	1	4	6	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	5	5	9	0
G0164	1	32	0	2	13	14	1	1	3	1	4	6	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	6	6	9	0
G0171	1	34	0	2	16	16	1	1	4	1	5	7	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	5	6	10	0
G0765	1	35	0	2	16	14	1	1	3	1	4	6	1	8	2	1	11	2	2	6	0	0	0	0	0	1	1	6	5	9	0
G1043	1	32	0	2	13	15	1	1	4	1	5	6	1	10	2	1	11	2	2	7	0	0	0	0	0	1	1	5	5	10	0
G2197	1	36	0	2	16	13	1	1	4	1	4	7	1	10	2	1	10	2	2	6	0	0	0	0	0	1	1	6	6	8	0
G2427	1	38	0	2	16	13	1	1	4	1	5	6	1	8	2	1	9	2	2	6	0	0	0	0	0	1	1	6	6	9	0
G2428	1	33	0	2	13	15	1	1	3	1	4	6	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	6	5	9	0
G2517	1	33	0	2	16	13	1	1	3	1	5	6	1	8	2	1	11	2	2	7	0	0	0	0	0	1	1	6	5	10	0
G2978	1	34	0	2	13	15	1	1	3	1	4	6	1	10	2	1	10	2	2	6	0	0	0	0	0	1	1	6	5	10	0
G3063	1	38	0	2	14	13	1	1	4	1	4	7	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	6	5	9	0
G3131	1	35	0	2	13	14	1	1	4	1	5	6	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	5	5	8	0

Can- order Level (c)	G 02 13	G 01 64	G 01 71	G 07 65	G 10 43	G 21 97	G 24 27	G 24 28	G 25 17	G 29 78	G 30 63	G 31 31	G 31 35	G 31 42	G 37 75	G 42 48	G 43 91	G 00 82	G 00 89	G 01 00	G 00 73	G 00 94	G 00 97	G 00 99	G 01 11	G 00 96	G 00 97	G 00 89	G 13 76	G 17 62	G 03 73
G3135	1	37	0	2	14	16	1	1	3	1	4	7	1	9	2	1	10	2	2	6	0	0	0	0	0	1	1	5	6	8	0
G3142	1	38	0	2	15	14	1	1	4	1	5	7	1	8	2	1	11	2	2	6	0	0	0	0	0	1	1	5	6	8	0
G3775	1	36	0	2	14	15	1	1	4	1	4	6	1	8	2	1	9	2	2	7	0	0	0	0	0	1	1	6	5	10	0
G4248	1	32	0	2	13	14	1	1	4	1	4	7	1	9	2	1	11	2	2	7	0	0	0	0	0	1	1	6	5	9	0
G4391	1	37	0	2	16	14	1	1	4	1	5	6	1	10	2	1	9	2	2	6	0	0	0	0	0	1	1	6	5	10	0
G0082	1	38	0	2	16	13	1	1	3	1	4	6	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	6	5	9	0
G0089	1	33	0	2	16	16	1	1	3	1	5	6	1	8	2	1	9	2	2	6	0	0	0	0	0	1	1	5	6	8	0
G0100	1	34	0	2	13	14	1	1	4	1	5	6	1	9	2	1	10	2	2	6	0	0	0	0	0	1	1	6	6	10	0
G0073	1	33	0	2	15	16	1	1	4	1	5	7	1	9	2	1	9	2	2	6	0	0	0	0	0	1	1	5	6	9	0
G0094	1	33	0	2	14	14	1	1	3	1	4	7	1	8	2	1	9	2	2	6	0	0	0	0	0	1	1	6	6	10	0
G0097	1	34	0	2	16	15	1	1	4	1	4	7	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	6	5	9	0
G0099	1	37	0	2	15	13	1	1	3	1	5	7	1	9	2	1	11	2	2	6	0	0	0	0	0	1	1	5	6	10	0
G0111	1	32	0	2	15	15	1	1	4	1	4	6	1	9	2	1	11	2	2	7	0	0	0	0	0	1	1	5	5	10	0
G0096	1	35	0	2	15	13	1	1	4	1	4	7	1	10	2	1	10	2	2	6	0	0	0	0	0	1	1	6	6	8	0
G0097	1	36	0	2	13	15	1	1	3	1	4	7	1	8	2	1	11	2	2	6	0	0	0	0	0	1	1	5	6	10	0
G0089	1	35	0	2	14	14	1	1	4	1	5	7	1	8	2	1	11	2	2	7	0	0	0	0	0	1	1	5	5	10	0
G1376	1	36	0	2	15	14	1	1	3	1	5	6	1	10	2	1	9	2	2	6	0	0	0	0	0	1	1	5	5	10	0
G1762	1	34	0	2	16	14	1	1	4	1	5	6	1	8	2	1	11	2	2	6	0	0	0	0	0	1	1	6	5	8	0
G0373	1	36	0	2	13	16	1	1	3	1	5	7	1	10	2	1	9	2	2	7	0	0	0	0	0	1	1	5	6	9	0

Hasil Order-up-to Level (S)

<i>Order-up-to Level (S)</i>	A0 116	A0 391	A0 408	A0 417	A0 419	A0 510	A0 531	A0 533	A0 650	A0 668	A0 812	A1 025	A1 216	A0 022	A1 857	A1 860	A1 876	A1 910	A2 276	A2 300	A2 319	A2 322	A2 330	A2 753	A0 185	A0 704
A0116	2	13	6	16	0	1	2	19	4	4	1	3	7	7	1	5	1	23	11	1	4	1	2	2	9	1
A0391	6	10	6	16	0	0	2	15	2	4	0	2	7	6	4	3	1	10	12	1	4	1	2	3	5	0
A0408	7	20	6	35	0	0	2	20	3	4	1	1	8	0	2	2	1	10	11	1	4	1	2	3	4	0
A0417	10	14	6	54	0	0	2	13	3	4	1	1	7	7	3	4	1	7	8	1	4	1	2	2	5	0
A0419	2	13	7	18	1	1	2	14	5	4	1	7	6	7	0	3	1	7	11	1	4	1	2	1	9	1
A0510	3	21	6	58	1	0	2	14	2	4	1	6	6	4	0	3	1	28	9	1	4	1	2	2	10	0
A0531	3	14	7	19	1	0	2	21	3	4	0	8	7	1	3	4	1	20	6	1	4	1	2	4	12	0
A0533	12	10	7	36	0	0	2	21	3	4	0	3	8	6	3	5	1	24	6	1	4	1	2	2	9	0
A0650	6	13	5	55	0	0	2	15	3	4	1	0	8	7	3	3	1	19	7	1	4	1	2	4	5	1
A0668	5	24	7	21	0	1	2	17	9	4	0	6	8	7	4	5	1	23	7	1	4	1	2	1	8	1
A0812	7	13	6	15	0	0	2	16	4	4	0	8	8	6	3	5	1	19	10	1	4	1	2	3	3	0
A1025	3	23	5	13	1	1	2	21	4	4	1	6	5	8	1	3	1	12	11	1	4	1	2	4	3	1
A1216	7	22	6	27	1	0	2	16	5	4	1	3	5	1	2	4	1	16	8	1	4	1	2	3	7	1
A0022	11	16	7	42	0	0	2	18	6	4	0	2	8	2	2	5	1	10	9	1	4	1	2	2	12	1

Order-up-to Level (S)	A0 116	A0 391	A0 408	A0 417	A0 419	A0 510	A0 531	A0 533	A0 650	A0 668	A0 812	A1 025	A1 216	A0 022	A1 857	A1 860	A1 876	A1 910	A2 276	A2 300	A2 319	A2 322	A2 330	A2 753	A0 185	A0 704
A1857	12	12	5	35	0	1	2	23	8	4	0	7	7	6	0	5	1	14	8	1	4	1	2	0	1	0
A1860	4	11	7	18	1	1	2	14	7	4	0	0	8	7	1	2	1	14	9	1	4	1	2	4	6	1
A1876	6	23	7	12	0	1	2	12	9	4	1	4	8	7	2	4	1	28	7	1	4	1	2	1	1	1
A1910	12	14	7	52	1	1	2	24	2	4	0	5	7	2	3	5	1	8	7	1	4	1	2	3	4	0
A2276	12	10	7	44	0	1	2	17	9	4	1	2	7	0	4	4	1	21	6	1	4	1	2	3	13	0
A2300	0	20	5	15	1	1	2	22	6	4	1	0	5	0	4	4	1	9	8	1	4	1	2	2	6	0
A2319	0	16	7	30	1	1	2	15	4	4	0	2	8	8	2	5	1	14	12	1	4	1	2	2	12	1
A2322	12	19	6	30	0	0	2	24	3	4	0	2	6	0	4	5	1	9	7	1	4	1	2	4	4	0
A2330	4	15	7	13	0	0	2	19	7	4	1	6	8	0	0	4	1	26	10	1	4	1	2	1	11	1
A2753	13	11	6	60	1	1	2	15	6	4	1	5	7	7	2	5	1	5	9	1	4	1	2	4	11	1
A0185	8	15	7	49	1	0	2	11	2	4	1	6	7	3	3	5	1	13	11	1	4	1	2	0	8	1
A0704	9	16	7	50	0	0	2	16	6	4	1	0	8	5	4	5	1	9	6	1	4	1	2	1	4	1
A0705	1	18	7	43	1	1	2	12	7	4	1	7	7	0	0	2	1	6	10	1	4	1	2	2	2	0
A0742	6	14	6	49	1	0	2	14	4	4	0	5	6	6	4	3	1	21	10	1	4	1	2	1	0	1
A0753	10	16	7	46	1	1	2	19	3	4	1	2	6	0	4	2	1	19	11	1	4	1	2	2	8	1
A0099	6	18	7	36	1	0	2	18	3	4	0	3	8	8	1	2	1	6	5	1	4	1	2	3	12	0
A2230	8	21	6	21	1	0	2	15	6	4	0	0	5	2	0	5	1	10	11	1	4	1	2	2	14	0
A2600	13	22	5	48	1	1	2	20	8	4	1	2	6	4	3	5	1	18	11	1	4	1	2	4	0	0
A3336	11	22	7	45	1	0	2	15	9	4	0	3	7	3	1	3	1	25	7	1	4	1	2	4	8	1
A4311	1	16	6	52	0	0	2	21	8	4	0	2	7	4	0	3	1	7	7	1	4	1	2	2	0	1
A0135	2	12	7	26	1	1	2	16	4	4	1	2	7	6	4	5	1	23	10	1	4	1	2	3	9	0
A0144	5	17	5	42	0	1	2	23	8	4	0	5	8	4	3	2	1	21	8	1	4	1	2	0	13	1
A0147	9	19	7	39	0	1	2	14	2	4	0	4	7	5	4	4	1	9	10	1	4	1	2	4	13	1
A0150	10	13	7	36	1	0	2	14	2	4	1	6	6	0	3	2	1	7	6	1	4	1	2	0	13	0
A0074	3	23	6	20	0	1	2	23	4	4	0	8	6	4	2	4	1	6	10	1	4	1	2	4	5	1
A0120	5	13	6	6	0	1	2	12	3	4	1	7	7	8	2	5	1	7	10	1	4	1	2	0	4	0
A0134	5	18	6	56	1	0	2	19	8	4	0	2	8	5	3	5	1	9	12	1	4	1	2	0	8	1
A0158	12	23	5	29	0	1	2	24	6	4	1	8	7	4	2	4	1	15	6	1	4	1	2	2	13	1
A0161	4	23	7	58	1	1	2	24	5	4	0	8	6	6	4	4	1	27	11	1	4	1	2	0	2	0
A0162	13	15	5	10	1	0	2	12	9	4	1	1	6	5	4	3	1	5	6	1	4	1	2	1	14	1
A0171	12	17	6	39	0	0	2	24	2	4	0	3	7	5	4	4	1	12	7	1	4	1	2	2	11	0
A0176	9	24	7	60	0	1	2	19	7	4	1	4	7	0	0	3	1	26	12	1	4	1	2	1	2	1
A0060	11	20	7	12	0	1	2	23	6	4	1	2	7	2	4	4	1	24	7	1	4	1	2	0	11	0
A0154	9	12	6	59	1	0	2	17	6	4	0	7	7	3	3	5	1	6	9	1	4	1	2	1	1	1
A0237	3	13	7	53	0	1	2	15	2	4	0	8	7	5	3	2	1	7	6	1	4	1	2	4	8	0
A0090	10	10	6	24	0	0	2	13	3	4	1	6	8	3	0	2	1	10	7	1	4	1	2	1	3	1
A0618	0	15	7	36	1	0	2	23	3	4	1	7	6	7	2	4	1	19	12	1	4	1	2	3	13	0
A0934	0	17	5	16	1	0	2	19	9	4	0	2	7	1	1	2	1	10	7	1	4	1	2	4	4	0

Order-up-to Level (S)	A0 705	A0 742	A0 753	A0 099	A2 230	A2 600	A3 336	A4 311	A0 135	A0 144	A0 147	A0 150	A0 074	A0 120	A0 134	A0 158	A0 161	A0 162	A0 171	A0 176	A0 060	A0 154	A0 237	A0 090	A0 618	A0 934
A0116	0	1	2	2	12	1	1	150	0	15	1	2	1	6	1	1	8	1	1	3	2	9	2	2	1	17
A0391	2	1	2	2	27	2	1	28	0	22	0	1	2	6	1	0	8	1	1	3	0	15	3	2	11	0
A0408	3	0	2	2	7	7	1	48	1	7	0	1	0	5	1	0	6	1	1	2	1	11	2	2	13	1
A0417	1	0	2	1	16	2	1	152	1	65	0	0	2	5	1	0	8	1	1	2	2	13	3	2	30	13
A0419	2	1	2	2	17	8	1	86	0	39	2	1	2	6	1	0	7	1	1	2	0	12	2	2	4	16
A0510	2	1	2	1	20	1	1	106	0	42	2	1	0	4	1	1	8	1	1	3	0	18	2	2	5	16
A0531	0	1	2	1	21	8	1	70	1	19	2	0	2	5	1	0	7	1	1	3	1	16	2	2	17	4
A0533	0	0	2	1	14	5	1	158	1	36	1	1	2	5	1	1	8	1	1	2	0	12	2	2	12	3
A0650	1	0	2	2	22	4	1	119	0	40	2	1	2	6	1	0	7	1	1	3	1	13	3	2	14	2
A0668	2	0	2	2	27	3	1	98	0	31	1	2	2	5	1	1	8	1	1	3	1	16	2	2	28	6
A0812	1	1	2	1	27	5	1	55	0	47	2	2	2	5	1	1	9	1	1	3	1	11	2	2	32	15
A1025	3	0	2	1	15	3	1	92	0	35	2	2	2	6	1	0	8	1	1	3	2	11	3	2	29	13
A1216	1	0	2	2	29	4	1	135	1	60	2	0	0	4	1	1	7	1	1	3	1	13	3	2	0	1
A0022	1	0	2	1	17	1	1	146	1	43	1	2	0	5	1	1	8	1	1	3	0	16	2	2	13	8
A1857	1	0	2	1	16	2	1	103	1	59	0	2	0	6	1	0	9	1	1	3	0	14	2	2	10	9
A1860	1	0	2	2	26	6	1	23	0	13	2	2	2	5	1	0	7	1	1	2	0	18	3	2	6	2
A1876	3	1	2	2	19	7	1	47	0	15	0	1	1	5	1	0	8	1	1	3	1	13	2	2	29	8
A1910	1	0	2	2	31	3	1	43	1	17	2	1	0	6	1	0	9	1	1	3	1	18	2	2	25	0
A2276	0	1	2	2	27	3	1	159	1	24	0	0	2	6	1	0	6	1	1	3	1	17	3	2	22	12
A2300	2	1	2	1	22	6	1	100	0	35	0	1	1	6	1	1	9	1	1	2	1	17	3	2	1	18
A2319	1	0	2	1	9	5	1	132	1	33	0	1	1	6	1	1	7	1	1	3	0	17	3	2	30	16
A2322	2	0	2	2	30	7	1	144	1	23	0	2	2	5	1	0	9	1	1	2	1	14	2	2	28	14
A2330	3	1	2	1	20	3	1	113	0	31	1	2	0	6	1	1	9	1	1	3	2	16	2	2	23	1
A2753	2	0	2	2	23	2	1	145	0	42	0	1	2	5	1	0	8	1	1	2	0	15	2	2	9	15
A0185	0	0	2	1	21	1	1	15	0	39	2	0	1	5	1	0	9	1	1	3	2	17	2	2	29	11
A0704	0	1	2	2	20	5	1	50	0	41	1	0	2	5	1	1	9	1	1	3	2	18	3	2	5	9
A0705	2	0	2	2	17	5	1	61	1	59	2	0	1	6	1	1	7	1	1	2	2	12	3	2	18	12
A0742	0	0	2	1	8	5	1	69	0	19	2	2	1	5	1	1	9	1	1	2	2	11	3	2	11	3
A0753	0	0	2	1	25	8	1	65	1	23	0	0	0	6	1	1	7	1	1	3	0	15	3	2	12	12
A0099	3	1	2	1	11	6	1	90	1	16	1	0	1	5	1	0	7	1	1	2	1	15	3	2	22	12
A2230	1	0	2	1	25	2	1	76	1	46	0	2	2	6	1	0	8	1	1	2	2	13	3	2	18	13
A2600	2	1	2	1	12	6	1	154	1	67	1	1	1	5	1	0	8	1	1	2	1	18	2	2	8	0
A3336	0	1	2	1	26	5	1	73	1	48	2	2	0	6	1	0	7	1	1	2	2	15	3	2	7	3
A4311	3	0	2	2	19	7	1	106	0	12	1	0	0	6	1	0	7	1	1	3	0	11	3	2	9	16
A0135	1	0	2	1	20	6	1	133	0	46	1	0	1	6	1	0	8	1	1	3	2	10	3	2	14	6
A0144	3	0	2	1	23	4	1	88	1	38	2	0	1	6	1	0	7	1	1	3	1	11	3	2	19	18
A0147	1	0	2	2	26	8	1	42	0	63	2	0	1	6	1	0	9	1	1	3	2	17	3	2	15	8
A0150	2	0	2	2	20	3	1	136	1	21	2	1	2	5	1	1	9	1	1	3	1	13	3	2	9	14
A0074	3	1	2	2	16	1	1	98	0	24	1	2	2	5	1	1	6	1	1	2	0	14	3	2	26	19
A0120	0	0	2	2	27	6	1	155	0	58	0	0	1	6	1	0	7	1	1	2	2	17	3	2	17	2
A0134	1	0	2	2	15	3	1	102	1	62	2	0	2	5	1	1	7	1	1	3	1	10	3	2	25	19

<i>Order-up-to Level (S)</i>	A0 705	A0 742	A0 753	A0 099	A2 230	A2 600	A3 336	A4 311	A0 135	A0 144	A0 147	A0 150	A0 074	A0 120	A0 134	A0 158	A0 161	A0 162	A0 171	A0 176	A0 060	A0 154	A0 237	A0 090	A0 618	A0 934
A0158	2	0	2	2	24	3	1	74	1	32	2	2	1	5	1	0	8	1	1	2	0	18	3	2	6	2
A0161	3	1	2	1	8	3	1	61	1	11	2	2	1	6	1	1	9	1	1	3	2	13	3	2	27	17
A0162	1	0	2	2	25	2	1	84	0	38	1	0	1	6	1	0	8	1	1	2	2	12	2	2	12	7
A0171	2	1	2	1	28	2	1	75	1	50	1	0	0	6	1	1	8	1	1	3	1	16	3	2	16	17
A0176	1	0	2	2	12	5	1	113	0	69	1	1	2	4	1	0	7	1	1	3	0	15	3	2	31	11
A0060	3	1	2	1	19	3	1	38	1	59	1	2	1	4	1	1	7	1	1	3	0	16	3	2	1	17
A0154	2	1	2	1	26	7	1	149	1	34	0	1	0	5	1	1	7	1	1	2	2	13	3	2	17	12
A0237	2	1	2	2	12	7	1	149	0	70	2	2	1	5	1	0	9	1	1	2	0	18	2	2	3	0
A0090	3	0	2	2	17	4	1	126	0	50	1	2	2	4	1	1	7	1	1	3	2	11	3	2	12	15
A0618	3	0	2	2	30	6	1	67	1	50	0	2	2	6	1	0	8	1	1	3	0	18	3	2	17	7
A0934	3	1	2	1	8	4	1	103	0	35	2	1	2	6	1	1	6	1	1	3	2	17	3	2	24	16

<i>Order-up-to Level (S)</i>	C08 18	C08 22	C08 23	C08 48	C13 14	C13 62	C13 64	C13 66	C13 69	C13 82	C02 06	C09 95	C14 86	C14 88	C22 11	C22 26	C24 22	C24 25	C25 03	C02 55	C02 59	C00 61
C0818	5	7	2	2	2	1	10	4	7	1	17	1	1	1	3	11	0	2	2	12	48	2
C0822	4	6	2	4	3	1	6	4	7	1	7	1	1	1	2	10	0	0	0	10	75	0
C0823	4	4	4	4	0	1	10	4	8	1	36	1	1	1	2	13	1	0	1	9	31	1
C0848	6	5	3	2	1	1	9	3	6	1	13	1	1	1	3	12	1	2	1	12	15	0
C1314	7	4	6	3	4	1	9	4	7	1	25	1	1	1	4	14	0	0	2	18	70	1
C1362	7	5	6	4	0	1	8	4	7	1	13	1	1	1	2	14	0	1	1	19	23	2
C1364	5	5	4	4	2	1	10	3	6	1	18	1	1	1	2	10	1	0	1	12	62	1
C1366	5	7	3	2	1	1	7	3	8	1	19	1	1	1	4	14	0	1	1	18	64	0
C1369	6	6	7	2	2	1	9	4	5	1	25	1	1	1	4	8	0	2	0	9	74	2
C1382	6	7	5	3	3	1	8	4	7	1	14	1	1	1	2	11	1	1	2	15	58	2
C0206	7	7	5	3	3	1	7	4	6	1	30	1	1	1	2	12	1	2	2	18	48	1
C0995	5	4	2	3	3	1	10	4	8	1	17	1	1	1	4	10	1	0	2	14	63	0
C1486	5	5	3	4	0	1	8	4	8	1	4	1	1	1	3	10	1	2	0	18	73	1
C1488	4	3	4	3	4	1	7	4	7	1	32	1	1	1	3	12	0	2	0	15	23	0
C2211	5	6	2	4	3	1	6	4	6	1	35	1	1	1	4	14	0	2	2	16	38	2
C2226	4	5	5	4	2	1	8	4	7	1	15	1	1	1	2	6	0	2	0	16	40	2
C2422	6	4	7	4	3	1	9	3	8	1	5	1	1	1	4	10	0	0	0	11	74	0
C2425	6	4	7	4	0	1	8	4	7	1	13	1	1	1	2	9	0	2	2	9	57	1
C2503	5	7	2	4	2	1	10	4	7	1	17	1	1	1	3	10	1	1	1	10	44	1
C0255	5	4	6	4	0	1	10	4	8	1	5	1	1	1	4	13	0	1	1	19	41	2
C0259	6	4	6	2	0	1	10	4	8	1	13	1	1	1	3	14	1	1	1	12	30	2
C0061	5	4	4	2	2	1	9	4	8	1	16	1	1	1	2	10	1	0	2	14	16	0
C0002	4	6	2	4	4	1	7	4	6	1	12	1	1	1	3	8	0	0	0	8	22	2
C1243	4	7	6	3	0	1	10	4	8	1	19	1	1	1	4	11	1	2	1	19	75	1
C1945	7	4	5	2	4	1	7	4	5	1	31	1	1	1	2	9	0	1	0	13	33	0
C2759	5	7	4	2	4	1	10	4	6	1	31	1	1	1	3	10	1	0	1	11	15	1

<i>Order-up-to Level (S)</i>	C08 18	C08 22	C08 23	C08 48	C13 14	C13 62	C13 64	C13 66	C13 69	C13 82	C02 06	C09 95	C14 86	C14 88	C22 11	C22 26	C24 22	C24 25	C25 03	C02 55	C02 59	C00 61
C2845	7	3	4	4	4	1	7	4	7	1	33	1	1	1	2	14	1	0	2	17	38	2
C2871	6	4	4	4	2	1	7	3	7	1	34	1	1	1	2	10	1	2	0	16	29	0
C3079	6	5	4	2	2	1	8	3	7	1	23	1	1	1	2	7	0	1	1	15	36	1
C3289	6	5	3	3	4	1	10	4	7	1	37	1	1	1	2	7	1	1	2	10	61	0
C3462	4	7	3	3	2	1	9	3	7	1	19	1	1	1	3	8	0	1	2	19	25	2
C3613	6	6	3	2	0	1	9	4	8	1	28	1	1	1	3	13	0	2	1	17	24	2
C3640	4	6	7	3	3	1	8	4	8	1	35	1	1	1	3	11	1	0	2	13	31	1
C4119	7	5	3	3	2	1	10	4	5	1	16	1	1	1	2	12	0	2	1	19	40	2
C0072	4	3	6	3	0	1	7	3	7	1	7	1	1	1	3	14	1	1	0	8	19	1
C0273	5	4	4	4	4	1	9	4	7	1	14	1	1	1	2	14	1	1	1	13	65	0
C0018	4	5	3	4	0	1	10	4	8	1	13	1	1	1	4	11	0	2	0	16	61	1
C0075	4	7	3	2	3	1	9	4	8	1	16	1	1	1	2	12	1	0	1	8	61	0
C0269	4	4	2	3	0	1	8	4	8	1	15	1	1	1	3	6	1	1	1	12	51	2
C0281	5	5	5	4	2	1	7	4	7	1	21	1	1	1	2	12	1	1	1	15	18	0
C0035	4	6	7	4	3	1	6	4	8	1	21	1	1	1	2	10	0	2	2	8	66	0
C0074	5	6	5	3	3	1	7	4	8	1	20	1	1	1	3	6	0	2	2	15	32	1
C0225	7	6	3	4	3	1	6	4	8	1	21	1	1	1	4	11	1	2	2	15	23	2

<i>Order-up-to Level (S)</i>	C000 2	C124 3	C194 5	C275 9	C284 5	C287 1	C307 9	C328 9	C346 2	C361 3	C364 0	C411 9	C007 2	C027 3	C001 8	C007 5	C026 9	C028 1	C003 5	C007 4	C022 5
C0818	4	3	0	0	13	11	3	0	16	1	13	2	0	4	1	27	1	190	1	0	1
C0822	4	5	0	1	20	5	3	0	18	0	36	8	0	1	2	37	6	123	1	1	1
C0823	5	3	1	1	15	5	2	0	8	0	12	2	0	12	2	31	0	36	1	0	1
C0848	3	4	1	0	6	7	3	2	20	0	42	8	1	10	2	18	8	144	1	1	1
C1314	2	5	0	1	2	6	3	2	25	0	24	8	0	6	2	23	1	65	0	1	1
C1362	4	4	1	1	22	4	4	0	5	0	5	1	0	11	1	37	2	202	1	1	1
C1364	5	3	0	0	2	2	4	0	39	1	44	3	0	5	2	35	8	135	1	0	1
C1366	2	5	1	0	6	7	3	2	13	0	30	8	0	5	2	29	3	160	0	1	1
C1369	3	5	1	1	12	7	2	0	34	0	24	5	1	7	1	36	0	274	0	0	1
C1382	5	5	0	0	12	10	4	2	20	0	22	2	0	1	1	36	8	209	0	1	1
C0206	5	5	1	1	17	5	3	0	19	1	9	0	0	14	2	35	8	77	0	0	1
C0995	5	5	0	1	8	10	4	2	15	1	31	1	0	9	1	26	8	105	0	1	1
C1486	4	4	0	1	14	10	4	0	37	1	45	4	0	1	1	35	8	181	0	1	1
C1488	3	4	0	1	7	6	4	0	2	1	41	4	1	0	2	27	7	89	1	1	1
C2211	3	5	0	1	6	11	3	0	23	0	14	4	0	3	1	30	1	338	1	1	1
C2226	4	4	0	0	7	1	2	2	26	0	42	8	1	5	2	27	8	219	1	1	1
C2422	3	3	1	1	15	6	4	0	8	1	15	5	0	9	1	33	3	217	0	1	1
C2425	2	4	1	1	19	4	4	2	0	1	34	1	1	10	1	38	9	178	1	0	1
C2503	3	5	1	1	0	11	4	0	1	0	36	2	0	0	2	31	9	144	1	1	1
C0255	5	3	1	1	0	12	4	1	18	0	14	2	0	6	1	38	8	210	1	0	1

<i>Order-up-to Level (S)</i>	C000 2	C124 3	C194 5	C275 9	C284 5	C287 1	C307 9	C328 9	C346 2	C361 3	C364 0	C411 9	C007 2	C027 3	C001 8	C007 5	C026 9	C028 1	C003 5	C007 4	C022 5
C0259	4	4	1	0	5	11	4	1	35	0	35	0	1	1	1	34	8	329	0	0	1
C0061	5	5	1	1	10	12	4	2	6	0	31	3	0	5	1	22	6	279	0	0	1
C0002	2	4	0	0	18	10	2	0	37	1	12	5	0	3	2	29	8	288	1	1	1
C1243	2	3	0	0	10	9	4	2	34	1	44	0	0	6	2	18	9	287	1	1	1
C1945	2	4	0	1	20	7	4	0	19	1	45	9	1	8	1	18	3	182	1	1	1
C2759	5	5	0	0	5	10	4	0	37	0	30	6	0	15	2	23	7	56	1	1	1
C2845	5	3	0	0	21	10	3	1	3	1	40	6	1	15	1	37	1	73	0	1	1
C2871	3	4	1	1	19	3	2	0	15	0	16	7	0	2	1	22	7	144	1	1	1
C3079	5	3	1	0	13	7	2	2	27	1	15	2	1	6	1	28	6	141	1	1	1
C3289	4	5	1	1	19	3	2	0	25	1	25	8	1	11	2	30	2	165	1	1	1
C3462	4	5	1	1	16	5	4	1	19	0	12	5	1	13	1	32	3	277	1	1	1
C3613	5	3	1	0	4	2	2	1	3	1	15	7	0	3	1	28	8	256	0	1	1
C3640	4	5	1	0	20	1	4	2	9	1	15	2	0	10	2	23	5	257	0	1	1
C4119	5	5	0	1	16	1	3	2	23	0	10	9	0	0	1	28	1	320	0	0	1
C0072	4	5	0	1	12	0	2	0	4	0	36	7	0	14	1	29	1	318	0	0	1
C0273	4	4	1	1	11	10	2	0	7	1	33	1	1	9	2	22	3	49	1	0	1
C0018	4	5	1	1	10	0	4	1	1	0	9	5	0	9	1	25	7	159	0	1	1
C0075	4	4	1	1	13	0	3	1	17	0	32	6	1	0	1	17	4	110	0	1	1
C0269	4	4	0	0	22	2	2	1	23	1	27	5	1	1	2	20	8	282	0	0	1
C0281	3	5	1	1	4	9	3	2	15	0	23	2	1	6	1	36	5	246	0	1	1
C0035	5	5	0	1	12	2	2	1	2	1	14	2	0	5	2	21	0	152	1	1	1
C0074	4	4	1	0	11	12	4	0	21	1	1	3	1	13	2	34	1	104	0	0	1
C0225	2	4	1	0	20	11	2	0	22	1	29	4	1	13	1	33	1	96	0	1	1

<i>Order-up-to Level (S)</i>	G 12 03	G 12 18	G 12 95	G 12 97	G 02 86	G 04 72	G 04 73	G 04 75	G 04 76	G 05 95	G 05 96	G 05 97	G 05 98	G 05 99	G 07 32	G 07 33	G 07 34	G 08 35	G 10 95	G 18 78	G 20 02	G 20 04	G 23 43	G 24 15	G 02 88	G 00 23	G 00 62	G 01 09	G 01 10	G 01 81	G 02 12
G1203	1	4	15	10	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	11	9	1	1	16	1	1	3	2	3	1
G1218	1	1	14	14	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	23	1	1	2	3	3	1
G1295	1	4	16	11	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	17	1	1	3	3	3	1
G1297	1	1	12	12	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	11	12	1	1	11	1	1	4	3	3	1
G0286	1	1	10	7	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	11	11	1	1	23	1	1	3	2	3	1
G0472	1	4	16	15	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	17	1	1	2	3	3	1
G0473	1	1	13	11	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	11	9	1	1	22	1	1	4	3	3	1
G0475	1	0	10	11	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	9	9	1	1	23	1	1	2	2	3	1
G0476	1	4	11	9	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	8	12	1	1	15	1	1	2	2	3	1
G0595	1	4	13	13	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	5	1	1	3	3	3	1
G0596	1	2	12	7	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	21	1	1	4	3	3	1
G0597	1	3	9	8	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	15	1	1	2	3	3	1
G0598	1	3	16	9	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	5	1	1	2	3	3	1

Order-up-to Level (S)	G 12 03	G 12 18	G 12 95	G 12 97	G 02 86	G 04 72	G 04 73	G 04 75	G 04 76	G 05 95	G 05 96	G 05 97	G 05 98	G 05 99	G 07 32	G 07 33	G 07 34	G 08 35	G 10 95	G 18 78	G 20 02	G 20 04	G 23 43	G 24 15	G 02 88	G 00 23	G 00 62	G 01 09	G 01 10	G 01 81	G 02 12
G0599	1	3	14	8	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	12	9	1	1	14	1	1	2	2	3	1
G0732	1	3	10	15	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	12	12	1	1	12	1	1	4	2	3	1
G0733	1	3	13	11	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	11	10	1	1	18	1	1	2	2	3	1
G0734	1	4	13	14	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	9	1	1	2	2	3	1
G0835	1	4	15	16	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	11	1	1	4	2	3	1
G1095	1	4	16	9	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	9	9	1	1	17	1	1	4	2	3	1
G1878	1	1	10	9	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	12	1	1	4	2	3	1
G2002	1	3	12	10	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	9	12	1	1	10	1	1	2	3	3	1
G2004	1	3	10	8	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	9	1	1	18	1	1	2	2	3	1
G2343	1	0	11	13	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	11	1	1	4	2	3	1
G2415	1	4	16	12	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	9	1	1	4	2	3	1
G0288	1	4	16	10	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	11	10	1	1	5	1	1	4	3	3	1
G0023	1	1	10	13	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	11	10	1	1	11	1	1	2	2	3	1
G0062	1	0	9	16	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	19	1	1	3	3	3	1
G0109	1	4	14	14	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	16	1	1	3	2	3	1
G0110	1	4	11	9	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	19	1	1	3	2	3	1
G0181	1	2	16	11	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	10	1	1	3	2	3	1
G0212	1	0	15	9	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	11	11	1	1	23	1	1	3	2	3	1
G0213	1	0	12	13	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	9	12	1	1	23	1	1	4	2	3	1
G0164	1	1	13	13	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	15	1	1	4	2	3	1
G0171	1	0	15	13	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	18	1	1	2	3	3	1
G0765	1	3	16	16	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	9	10	1	1	13	1	1	4	3	3	1
G1043	1	0	13	16	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	9	1	1	3	2	3	1
G2197	1	0	11	9	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	11	12	1	1	12	1	1	2	2	3	1
G2427	1	2	13	9	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	8	12	1	1	13	1	1	2	2	3	1
G2428	1	4	11	15	1	1	2	1	1	1	1	1	1	2	1	1	4	1	2	1	11	9	1	1	11	1	1	2	2	3	1
G2517	1	4	11	8	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	11	10	1	1	9	1	1	3	3	3	1
G2978	1	4	15	12	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	14	1	1	4	2	3	1
G3063	1	1	9	14	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	10	11	1	1	6	1	1	4	3	3	1
G3131	1	1	14	14	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	14	1	1	3	3	3	1
G3135	1	2	14	9	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	11	10	1	1	21	1	1	3	2	3	1
G3142	1	1	13	7	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	17	1	1	2	3	3	1
G3775	1	1	16	15	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	11	12	1	1	24	1	1	4	2	3	1
G4248	1	1	14	9	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	9	10	1	1	10	1	1	3	3	3	1
G4391	1	0	11	13	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	10	1	1	3	3	3	1
G0082	1	3	16	11	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	12	11	1	1	22	1	1	3	3	3	1
G0089	1	4	16	11	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	12	12	1	1	24	1	1	2	2	3	1
G0100	1	0	15	15	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	22	1	1	4	3	3	1
G0073	1	4	12	9	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	11	1	1	4	2	3	1
G0094	1	4	9	9	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	20	1	1	2	3	3	1

Order-up-to Level (S)	G 12 03	G 12 18	G 12 95	G 12 97	G 02 86	G 04 72	G 04 73	G 04 75	G 04 76	G 05 95	G 05 96	G 05 97	G 05 98	G 05 99	G 07 32	G 07 33	G 07 34	G 08 35	G 10 95	G 18 78	G 20 02	G 20 04	G 23 43	G 24 15	G 02 88	G 00 23	G 00 62	G 01 09	G 01 10	G 01 81	G 02 12
G0097	1	1	12	8	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	10	1	1	3	3	3	1
G0099	1	2	9	14	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	11	12	1	1	23	1	1	4	2	3	1
G0111	1	3	15	15	1	1	2	0	0	1	1	1	1	2	1	1	4	1	2	1	10	10	1	1	7	1	1	3	2	3	1
G0096	1	2	14	9	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	9	12	1	1	8	1	1	2	2	3	1
G0097	1	0	11	13	1	1	2	0	1	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	9	1	1	3	3	3	1
G0089	1	1	9	14	1	1	2	2	1	1	1	1	1	2	1	1	4	1	2	1	10	12	1	1	23	1	1	4	2	3	1
G1376	1	1	13	14	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	12	9	1	1	20	1	1	3	2	3	1
G1762	1	1	10	11	1	1	2	2	0	1	1	1	1	2	1	1	4	1	2	1	11	9	1	1	17	1	1	2	2	3	1
G0373	1	1	8	13	1	1	2	1	0	1	1	1	1	2	1	1	4	1	2	1	12	10	1	1	9	1	1	3	3	3	1

Order-up-to Level (S)	G 0 2 1 3	G01 64	G0 171	G 0 7 6 5	G 10 43	G21 97	G 2 4 2 7	G 24 28	G 25 17	G 29 78	G 30 63	G 31 31	G 31 35	G 31 42	G 37 75	G 42 48	G 43 91	G 00 82	G 00 89	G 01 00	G 00 73	G 00 94	G 00 97	G 00 99	G 01 11	G 00 96	G 00 97	G 00 89	G 13 76	G 17 62	G 03 73
G1203	1	565	74	2	43	26	1	1	2	1	4	10	1	4	2	1	18	3	2	17	1	3	3	1	1	1	1	14	22	12	3
G1218	1	271	196	2	43	71	1	1	2	1	6	8	1	4	2	1	13	3	2	11	1	3	0	0	0	1	1	7	16	17	2
G1295	1	230	126	2	35	70	1	1	2	1	6	12	1	4	2	1	18	4	2	6	1	1	0	0	1	1	1	16	25	14	2
G1297	1	68	60	2	52	87	1	1	2	1	6	11	1	4	2	1	15	2	2	23	0	2	3	0	2	1	1	25	31	11	0
G0286	1	555	103	2	52	45	1	1	2	1	5	12	1	4	2	1	17	4	2	8	1	0	1	1	0	1	1	22	33	14	3
G0472	1	346	195	2	37	83	1	1	2	1	5	11	1	4	2	1	12	4	2	22	1	2	2	0	1	1	1	10	9	10	2
G0473	1	45	178	2	37	27	1	1	2	1	6	9	1	4	2	1	17	3	2	7	1	1	2	1	1	1	1	8	20	9	3
G0475	1	43	182	2	21	82	1	1	2	1	6	10	1	4	2	1	15	3	2	21	0	3	2	1	1	1	1	13	10	10	1
G0476	1	266	29	2	29	78	1	1	2	1	5	9	1	4	2	1	16	4	2	24	1	3	2	0	1	1	1	27	7	16	2
G0595	1	454	98	2	52	89	1	1	2	1	5	7	1	4	2	1	16	3	2	15	1	2	3	0	1	1	1	14	23	11	3
G0596	1	187	64	2	29	45	1	1	2	1	4	8	1	4	2	1	17	3	2	23	0	1	1	1	1	1	1	13	26	17	1
G0597	1	135	188	2	41	44	1	1	2	1	5	10	1	4	2	1	12	2	2	10	1	1	1	0	1	1	1	28	23	10	3
G0598	1	637	25	2	34	90	1	1	2	1	5	10	1	4	2	1	12	4	2	15	1	2	2	1	1	1	1	9	31	19	3
G0599	1	615	163	2	18	58	1	1	2	1	6	10	1	4	2	1	16	2	2	20	0	0	0	1	0	1	1	21	30	17	1
G0732	1	543	18	2	18	41	1	1	2	1	6	9	1	4	2	1	17	4	2	17	0	3	2	1	2	1	1	27	27	17	2
G0733	1	468	173	2	39	50	1	1	2	1	5	10	1	4	2	1	13	4	2	23	0	0	1	1	1	1	1	17	21	12	3
G0734	1	741	16	2	42	88	1	1	2	1	5	9	1	4	2	1	12	2	2	17	1	1	2	0	0	1	1	26	13	11	2
G0835	1	676	142	2	40	107	1	1	2	1	6	9	1	4	2	1	13	3	2	6	0	3	1	0	1	1	1	7	26	14	3
G1095	1	106	172	2	56	41	1	1	2	1	6	10	1	4	2	1	16	4	2	8	0	3	1	0	2	1	1	22	35	18	3
G1878	1	225	37	2	40	30	1	1	2	1	5	8	1	4	2	1	15	4	2	20	1	0	3	1	0	1	1	24	32	16	3
G2002	1	246	0	2	22	44	1	1	2	1	6	8	1	4	2	1	14	3	2	15	0	0	3	1	0	1	1	15	25	12	1
G2004	1	599	186	2	42	71	1	1	2	1	5	9	1	4	2	1	14	3	2	18	0	1	1	0	1	1	1	6	26	8	2
G2343	1	589	148	2	50	77	1	1	2	1	6	8	1	4	2	1	10	4	2	21	0	2	0	1	2	1	1	15	29	9	3
G2415	1	307	136	2	26	79	1	1	2	1	5	10	1	4	2	1	12	3	2	21	1	2	3	1	0	1	1	25	19	15	0

<i>Order- up-to Level (S)</i>	G 0 2 1 3	G01 64	G0 171	G 0 7 6 5	G 10 43	G21 97	G 2 4 2 7	G 24 28	G 25 17	G 29 78	G 30 63	G 31 31	G 31 35	G 31 42	G 37 75	G 42 48	G 43 91	G 00 82	G 00 89	G 01 00	G 00 73	G 00 94	G 00 97	G 00 99	G 01 11	G 00 96	G 00 97	G 00 89	G 13 76	G 17 62	G 03 73
G0288	1	421	187	2	30	100	1	1	2	1	4	11	1	4	2	1	11	2	2	14	0	0	3	1	0	1	1	20	26	16	2
G0023	1	733	116	2	50	69	1	1	2	1	4	11	1	4	2	1	12	2	2	7	0	2	0	0	0	1	1	18	20	14	2
G0062	1	79	143	2	18	76	1	1	2	1	5	11	1	4	2	1	13	4	2	11	0	3	2	0	0	1	1	11	28	14	0
G0109	1	288	115	2	20	25	1	1	2	1	6	9	1	4	2	1	16	3	2	18	0	2	2	0	2	1	1	23	19	13	2
G0110	1	530	34	2	51	61	1	1	2	1	6	12	1	4	2	1	12	4	2	10	0	2	3	0	0	1	1	19	25	15	2
G0181	1	183	102	2	25	31	1	1	2	1	6	8	1	4	2	1	15	3	2	9	1	0	0	1	2	1	1	17	24	10	0
G0212	1	547	189	2	28	27	1	1	2	1	6	9	1	4	2	1	11	2	2	9	0	0	3	0	1	1	1	13	35	18	2
G0213	1	628	74	2	51	38	1	1	2	1	5	7	1	4	2	1	11	2	2	20	0	0	1	1	1	1	1	20	9	10	2
G0164	1	127	192	2	26	107	1	1	2	1	5	12	1	4	2	1	17	4	2	21	1	1	1	0	1	1	1	25	7	12	0
G0171	1	394	145	2	56	34	1	1	2	1	6	8	1	4	2	1	16	3	2	19	0	0	0	1	1	1	1	28	23	15	0
G0765	1	687	171	2	40	85	1	1	2	1	4	7	1	4	2	1	11	3	2	15	0	0	1	1	0	1	1	14	8	13	0
G1043	1	118	83	2	45	86	1	1	2	1	6	9	1	4	2	1	15	2	2	14	1	2	0	1	1	1	1	22	27	11	0
G2197	1	511	12	2	55	108	1	1	2	1	6	8	1	4	2	1	16	4	2	21	1	0	2	1	0	1	1	25	35	11	2
G2427	1	471	179	2	28	37	1	1	2	1	6	8	1	4	2	1	11	4	2	10	0	3	3	0	0	1	1	22	15	10	1
G2428	1	598	181	2	27	24	1	1	2	1	5	10	1	4	2	1	18	2	2	9	0	3	3	0	2	1	1	25	27	15	3
G2517	1	194	21	2	41	62	1	1	2	1	6	12	1	4	2	1	18	2	2	15	0	2	2	0	0	1	1	9	10	16	1
G2978	1	513	84	2	23	51	1	1	2	1	6	9	1	4	2	1	14	3	2	14	1	2	2	0	1	1	1	20	17	15	3
G3063	1	664	173	2	24	15	1	1	2	1	6	12	1	4	2	1	13	4	2	18	1	1	2	0	1	1	1	6	35	15	2
G3131	1	360	198	2	38	79	1	1	2	1	5	7	1	4	2	1	14	4	2	18	0	1	1	1	0	1	1	9	15	16	1
G3135	1	478	108	2	39	19	1	1	2	1	6	10	1	4	2	1	15	2	2	11	1	0	3	0	1	1	1	25	6	8	2
G3142	1	158	35	2	62	19	1	1	2	1	5	9	1	4	2	1	13	2	2	11	0	3	0	1	1	1	1	7	31	9	1
G3775	1	239	126	2	16	68	1	1	2	1	5	9	1	4	2	1	14	4	2	24	0	1	3	0	2	1	1	6	18	18	1
G4248	1	440	63	2	32	106	1	1	2	1	6	12	1	4	2	1	13	2	2	15	0	3	0	0	1	1	1	16	23	10	3
G4391	1	730	131	2	39	33	1	1	2	1	6	10	1	4	2	1	11	4	2	9	1	0	0	0	2	1	1	8	24	13	3
G0082	1	362	14	2	24	39	1	1	2	1	5	10	1	4	2	1	9	2	2	10	1	0	2	1	2	1	1	24	9	17	1
G0089	1	181	74	2	22	88	1	1	2	1	5	10	1	4	2	1	14	3	2	22	1	2	0	0	0	1	1	19	25	17	0
G0100	1	480	140	2	43	97	1	1	2	1	5	8	1	4	2	1	15	4	2	21	1	1	3	1	0	1	1	21	17	19	0
G0073	1	690	57	2	30	76	1	1	2	1	5	10	1	4	2	1	15	2	2	20	0	3	3	1	1	1	1	24	10	18	0
G0094	1	529	34	2	39	88	1	1	2	1	4	7	1	4	2	1	11	4	2	15	0	0	3	1	0	1	1	25	9	17	1
G0097	1	396	122	2	25	23	1	1	2	1	5	11	1	4	2	1	17	2	2	11	0	3	1	1	2	1	1	28	15	10	2
G0099	1	570	123	2	62	83	1	1	2	1	5	8	1	4	2	1	15	4	2	18	0	1	3	1	0	1	1	6	23	10	1
G0111	1	714	135	2	47	94	1	1	2	1	6	12	1	4	2	1	13	3	2	14	0	0	1	0	2	1	1	11	18	19	1
G0096	1	357	40	2	49	64	1	1	2	1	4	9	1	4	2	1	13	3	2	12	0	3	2	0	1	1	1	11	7	18	1
G0097	1	369	104	2	42	24	1	1	2	1	5	8	1	4	2	1	12	3	2	8	0	0	0	1	0	1	1	12	13	13	2
G0089	1	555	77	2	28	53	1	1	2	1	6	8	1	4	2	1	15	3	2	13	0	3	0	0	2	1	1	22	25	12	1
G1376	1	438	97	2	58	106	1	1	2	1	5	6	1	4	2	1	13	2	2	10	1	1	1	0	1	1	1	12	35	18	2
G1762	1	597	185	2	60	88	1	1	2	1	6	10	1	4	2	1	18	2	2	15	0	0	0	0	2	1	1	19	22	16	2
G0373	1	597	98	2	54	56	1	1	2	1	5	9	1	4	2	1	16	4	2	12	1	3	2	0	0	1	1	13	35	12	1

Hasil Order Quantity (x)

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0116	0	14	0	0	0	0	0	0	0	0	0	0
A0391	10	0	0	0	0	0	24	0	0	0	0	0
A0408	10	0	0	1	0	0	0	2	0	0	0	0
A0417	0	0	0	73	0	0	0	0	0	0	0	0
A0419	1	0	0	0	0	0	0	0	1	0	0	0
A0510	0	0	0	1	0	0	0	0	0	0	0	0
A0531	2	0	0	0	0	0	0	2	0	0	0	0
A0533	21	0	0	0	0	0	24	0	0	0	0	0
A0650	6	1	0	0	4	1	0	0	0	0	0	0
A0668	4	0	0	0	0	4	0	0	0	0	0	0
A0812	0	0	0	0	0	0	1	0	0	0	0	0
A1025	6	0	0	0	0	0	4	0	0	0	0	0
A1216	11	0	0	0	2	0	0	0	0	0	0	0
A0022	0	0	0	2	0	4	2	0	0	0	0	0
A1857	0	0	0	4	0	0	0	0	0	0	0	0
A1860	6	0	1	0	0	0	0	0	0	0	0	0
A1876	0	0	0	1	1	0	0	0	0	0	0	0
A1910	8	0	0	20	0	8	0	0	0	0	0	0
A2276	4	0	0	9	3	0	0	0	0	2	0	0
A2300	0	0	0	0	0	0	0	0	1	1	0	0
A2319	4	0	0	0	0	2	0	0	0	0	2	0
A2322	0	0	0	0	1	0	0	1	0	0	0	0
A2330	0	0	0	0	0	2	0	2	0	0	0	0
A2753	4	0	0	0	0	0	0	0	0	0	0	0
A0185	15	0	0	0	0	0	0	0	0	0	0	0
A0704	1	0	1	0	0	0	0	0	0	0	0	0
A0705	3	0	0	0	0	0	0	0	0	0	0	0
A0742	0	0	1	0	0	0	0	0	0	0	0	0
A0753	0	2	2	0	0	0	0	0	0	0	0	0
A0099	1	0	1	0	0	1	0	0	0	0	0	0
A2230	24	0	0	0	0	0	0	0	0	15	15	0
A2600	5	0	0	9	0	0	0	0	0	0	0	0
A3336	0	0	0	0	1	0	0	1	0	0	0	0
A4311	175	0	0	0	0	0	0	0	0	0	0	0
A0135	0	1	0	0	0	0	0	0	0	0	0	0
A0144	37	0	0	0	33	0	5	0	5	0	0	0
A0147	0	2	0	0	0	0	0	0	0	0	0	0
A0150	0	3	0	0	0	0	0	0	0	0	0	0
A0074	2	0	0	0	0	0	0	0	0	1	0	0
A0120	0	0	4	0	0	2	0	4	0	0	2	0
A0134	0	1	1	0	0	0	0	0	0	0	0	0
A0158	0	0	0	1	0	0	0	0	0	0	0	0
A0161	3	0	0	5	1	6	0	3	0	0	0	0
A0162	0	1	0	1	0	0	0	0	0	0	0	0
A0171	1	0	0	1	0	0	0	0	0	0	0	0
A0176	2	4	0	0	0	0	0	0	0	0	0	0
A0060	0	0	2	0	0	0	0	0	0	0	0	0
A0154	11	0	8	12	0	0	0	0	0	0	0	0
A0237	1	0	2	2	0	0	0	0	0	0	0	0
A0090	0	0	0	0	0	1	0	2	0	0	0	0
A0618	17	0	0	0	0	0	16	0	0	0	0	0
A0934	0	0	0	0	0	0	0	0	35	0	0	0
C0818	5	0	0	0	0	0	6	0	0	0	0	0
C0822	6	0	0	3	3	0	0	0	0	0	0	0
C0823	4	0	3	3	0	0	0	0	0	0	0	0
C0848	2	0	0	0	2	0	2	0	0	0	0	0
C1314	4	0	0	0	0	0	0	0	0	0	0	0
C1362	0	0	0	1	1	0	0	0	0	0	0	0
C1364	0	4	0	6	6	4	0	0	0	0	0	0
C1366	0	3	1	1	2	0	0	0	0	0	0	0
C1369	5	0	0	0	0	8	0	0	1	0	0	0
C1382	0	0	0	0	0	0	0	0	1	0	1	0
C0206	29	0	0	8	4	4	0	0	0	0	0	0
C0995	1	0	1	0	0	0	0	0	0	0	0	0
C1486	1	0	1	0	0	0	0	0	0	0	0	0
C1488	1	0	1	0	0	0	0	0	0	0	0	0

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
C2211	4	0	0	0	0	4	0	0	0	0	0	0
C2226	12	6	2	0	0	0	0	0	0	0	0	0
C2422	0	0	1	0	0	0	0	0	0	0	0	0
C2425	2	0	0	0	1	0	0	0	0	0	0	0
C2503	1	0	1	0	0	0	0	0	0	0	0	0
C0255	18	0	1	7	7	0	0	0	0	0	0	0
C0259	30	0	0	0	46	11	11	0	0	0	0	0
C0061	0	0	2	0	0	0	0	0	0	0	0	0
C0002	2	0	0	0	1	0	3	0	0	0	1	0
C1243	3	0	0	0	1	1	0	2	0	0	1	0
C1945	0	0	0	0	0	0	1	0	0	0	0	0
C2759	0	0	1	0	0	0	0	0	0	0	0	0
C2845	0	0	0	0	0	0	0	0	24	0	0	0
C2871	4	3	0	0	5	0	0	0	0	0	0	0
C3079	0	6	0	0	0	0	0	0	0	0	0	0
C3289	0	0	0	0	2	0	0	0	0	0	0	0
C3462	0	0	0	0	19	0	40	0	0	0	0	0
C3613	1	1	0	0	0	0	0	0	0	0	0	0
C3640	30	0	0	0	30	0	0	0	0	0	0	0
C4119	10	0	0	0	0	0	0	0	0	0	0	0
C0072	0	0	1	0	0	0	0	0	0	0	0	0
C0273	3	0	12	0	6	0	0	0	0	0	0	0
C0018	1	1	0	0	1	0	0	0	0	0	0	0
C0075	19	0	8	12	0	4	8	0	0	0	4	0
C0269	9	0	0	0	0	0	0	0	0	0	0	0
C0281	370	0	0	0	0	0	0	0	0	0	0	0
C0035	1	0	1	0	0	0	0	0	0	0	0	0
C0074	0	1	0	0	0	0	0	0	0	0	0	0
C0225	1	0	0	0	0	0	1	0	0	0	0	0
G1203	1	0	0	0	0	0	1	0	0	0	0	0
G1218	1	0	0	0	0	0	0	4	0	0	0	0
G1295	12	0	0	0	0	5	11	0	0	0	0	0
G1297	8	0	0	0	0	8	10	0	0	0	0	0
G0286	1	0	0	1	0	0	0	0	0	0	0	0
G0472	1	0	0	0	0	0	1	0	0	0	0	0
G0473	2	0	0	0	0	0	2	0	0	0	0	0
G0475	1	0	0	2	0	0	0	0	0	0	0	0
G0476	0	0	0	1	0	0	0	0	0	0	0	0
G0595	1	0	0	1	0	0	0	0	0	0	0	0
G0596	1	0	0	1	0	0	0	0	0	0	0	0
G0597	1	0	0	1	0	0	0	0	0	0	0	0
G0598	1	0	0	1	0	0	0	0	0	0	0	0
G0599	2	0	0	2	0	0	0	0	0	0	0	0
G0732	1	0	0	1	0	0	0	0	0	0	0	0
G0733	1	0	0	1	0	0	0	0	0	0	0	0
G0734	4	0	0	4	0	0	0	0	0	0	0	0
G0835	0	0	2	0	0	0	0	0	0	0	0	0
G1095	2	0	0	0	0	0	2	0	0	0	0	0
G1878	1	0	0	0	1	0	0	0	0	0	0	0
G2002	9	0	0	12	0	0	0	0	0	0	0	0
G2004	9	0	0	12	0	0	0	0	0	0	0	0
G2343	1	0	0	1	0	0	0	0	0	0	0	0
G2415	1	0	0	0	0	0	1	0	0	0	0	0
G0288	4	0	0	9	1	0	3	5	3	2	2	5
G0023	0	0	0	0	0	0	0	2	0	0	0	0
G0062	0	0	0	1	1	0	0	0	0	0	0	0
G0109	2	0	0	0	1	1	2	0	0	0	1	0
G0110	1	0	0	0	0	1	2	0	0	0	1	0
G0181	3	0	0	3	0	0	0	0	0	0	0	0
G0212	1	0	0	0	0	0	1	0	0	0	0	0
G0213	1	0	0	0	0	0	1	0	0	0	0	0
G0164	127	752	0	0	0	0	0	0	0	0	0	0
G0171	345	0	0	0	0	0	0	0	0	0	0	0
G0765	2	0	0	0	0	0	2	0	0	0	0	0
G1043	58	0	0	29	0	0	0	0	0	0	0	0
G2197	144	0	0	0	0	0	0	0	0	0	0	0
G2427	1	0	0	0	0	0	1	0	0	0	0	0
G2428	1	0	0	0	0	0	1	0	0	0	0	0

Order Quantity	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
G2517	2	0	0	2	0	0	0	0	0	0	0	0
G2978	1	0	0	0	0	0	1	0	0	0	0	0
G3063	4	0	0	8	0	0	0	0	0	0	0	0
G3131	7	0	0	12	0	0	0	0	0	0	0	0
G3135	1	0	0	0	0	0	1	0	0	0	0	0
G3142	4	0	0	0	0	0	0	0	4	0	0	0
G3775	2	0	0	0	0	0	2	0	0	0	0	0
G4248	1	0	0	0	0	0	1	0	0	0	0	0
G4391	11	0	0	18	0	0	0	0	0	0	0	0
G0082	2	0	0	0	0	0	4	0	0	2	0	0
G0089	2	0	0	0	0	0	2	0	0	0	0	0
G0100	20	0	0	0	0	0	0	18	0	0	0	0
G0073	0	0	0	0	0	0	0	0	0	1	0	0
G0094	0	0	0	0	0	0	3	0	0	0	0	0
G0097	1	0	0	0	0	0	1	0	0	2	0	0
G0099	1	0	0	0	0	0	1	0	0	0	0	0
G0111	2	0	0	0	0	0	0	0	0	2	0	0
G0096	1	0	0	0	0	0	1	0	0	0	0	0
G0097	1	0	0	0	0	0	1	0	0	0	0	0
G0089	22	0	0	21	0	0	0	0	0	0	0	0
G1376	30	0	0	0	32	0	0	0	0	0	0	0
G1762	0	10	19	0	0	0	0	0	0	0	0	0
G0373	3	0	0	0	0	0	0	0	0	0	0	0

Hasil Level Inventory

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0116	1	1	1	1	1	1	1	1	1	1	1	1
A0391	10	10	10	10	10	10	10	10	10	10	10	10
A0408	6	6	6	6	6	6	4	6	6	6	6	6
A0417	8	8	8	32	31	30	29	29	29	13	13	13
A0419	1	1	1	1	1	1	1	1	1	1	1	1
A0510	0	0	0	0	0	0	0	0	0	0	0	0
A0531	2	2	2	2	2	0	0	2	2	2	2	2
A0533	21	21	21	21	21	21	21	21	21	21	21	21
A0650	3	3	3	0	2	3	3	3	3	3	3	3
A0668	4	4	4	4	4	4	4	4	4	4	4	4
A0812	1	1	1	0	0	0	0	0	0	0	0	0
A1025	6	4	4	4	2	2	4	2	2	2	2	2
A1216	7	7	7	3	5	5	5	5	5	5	5	5
A0022	2	2	2	2	2	0	0	0	0	0	0	0
A1857	0	0	0	0	0	0	0	0	0	0	0	0
A1860	2	2	2	2	2	2	2	2	2	2	2	2
A1876	1	1	1	0	1	1	1	1	1	1	1	1
A1910	8	8	0	8	5	8	8	7	6	6	6	6
A2276	6	6	0	7	6	6	6	4	4	6	6	6
A2300	1	1	1	1	1	0	0	0	0	1	1	1
A2319	4	4	4	4	4	4	4	4	2	2	4	4
A2322	1	1	1	1	1	1	1	1	1	1	1	1
A2330	2	2	2	2	0	2	2	2	2	2	2	2
A2753	4	4	4	4	4	0	0	0	0	0	0	0
A0185	1	1	1	1	1	1	1	1	1	1	1	1
A0704	1	1	1	1	1	1	1	1	1	1	1	1
A0705	0	0	0	0	0	0	0	0	0	0	0	0
A0742	0	0	0	0	0	0	0	0	0	0	0	0
A0753	0	2	2	2	2	2	2	2	2	2	2	2
A0099	0	0	1	0	0	1	1	1	1	1	1	1
A2230	25	25	25	24	24	16	0	0	0	15	22	22
A2600	5	5	5	6	6	6	6	6	6	6	6	6
A3336	1	1	1	1	1	1	1	1	1	1	1	1
A4311	101	87	73	59	45	31	15	15	15	15	15	15
A0135	1	0	0	0	0	0	0	0	0	0	0	0
A0144	38	38	27	27	0	0	5	5	10	10	10	10
A0147	0	0	0	0	0	0	0	0	0	0	0	0
A0150	2	1	1	1	1	1	1	1	1	1	1	1
A0074	2	2	2	2	2	1	1	1	1	1	1	1

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0120	2	2	6	6	6	6	2	6	6	6	6	6
A0134	0	0	1	1	1	1	1	1	1	1	1	1
A0158	1	1	1	0	0	0	0	0	0	0	0	0
A0161	9	7	5	9	0	6	6	9	9	9	9	9
A0162	1	1	1	1	1	1	1	1	1	1	1	1
A0171	1	1	1	1	1	1	1	1	1	1	1	1
A0176	3	3	3	3	3	3	3	3	3	3	3	3
A0060	2	2	0	0	0	0	0	0	0	0	0	0
A0154	13	9	13	13	13	13	13	13	13	13	13	13
A0237	2	2	0	2	2	2	2	2	2	2	2	2
A0090	3	3	3	3	2	2	0	2	2	2	1	1
A0618	17	1	1	1	1	1	1	1	1	1	1	1
A0934	19	19	19	19	19	19	19	19	16	16	16	16
C0818	5	5	5	5	3	1	4	4	4	4	4	4
C0822	6	5	5	6	6	5	5	5	5	5	5	5
C0823	4	4	4	4	4	4	4	4	4	3	3	3
C0848	2	2	2	2	2	2	2	2	2	2	2	2
C1314	4	4	4	4	4	4	0	0	0	0	0	0
C1362	1	1	1	1	1	1	1	1	1	1	1	1
C1364	6	10	10	10	10	10	10	10	10	10	10	10
C1366	0	3	4	3	3	3	3	3	3	3	3	3
C1369	5	5	5	5	5	5	5	5	6	6	6	6
C1382	5	5	5	3	3	1	1	1	1	0	1	1
C0206	30	21	1	0	4	8	8	8	8	8	8	8
C0995	1	1	1	1	1	1	1	1	1	1	1	1
C1486	1	1	1	1	1	1	1	1	1	1	1	1
C1488	1	1	1	1	1	1	1	1	1	1	1	1
C2211	4	4	4	4	4	4	4	4	4	4	4	4
C2226	6	6	6	6	6	6	6	6	6	6	6	6
C2422	0	0	0	0	0	0	0	0	0	0	0	0
C2425	1	1	1	1	1	1	1	1	1	1	1	1
C2503	1	0	0	0	0	0	0	0	0	0	0	0
C0255	19	19	0	7	14	14	14	14	14	14	14	14
C0259	30	30	30	30	0	11	22	22	22	22	22	22
C0061	1	1	0	0	0	0	0	0	0	0	0	0
C0002	2	2	2	2	2	1	2	2	2	2	2	2
C1243	3	3	3	3	3	3	1	3	3	3	3	3
C1945	1	1	1	1	1	1	0	0	0	0	0	0
C2759	0	0	1	0	0	0	0	0	0	0	0	0
C2845	1	1	1	1	1	1	1	1	1	1	1	1
C2871	0	3	3	3	0	0	0	0	0	0	0	0
C3079	2	2	2	2	2	2	2	2	2	2	2	2
C3289	0	0	0	0	0	0	0	0	0	0	0	0
C3462	40	40	40	40	19	19	19	19	19	19	19	19
C3613	1	1	1	1	1	1	1	1	1	1	1	1
C3640	15	15	15	15	15	15	15	15	15	15	15	15
C4119	9	9	9	1	1	1	1	1	1	1	1	1
C0072	0	0	0	0	0	0	0	0	0	0	0	0
C0273	0	0	6	6	6	6	6	6	6	6	6	6
C0018	0	1	1	1	1	1	1	1	1	1	1	1
C0075	17	13	17	17	17	17	17	17	17	17	17	17
C0269	0	0	0	0	0	0	0	0	0	0	0	0
C0281	31	31	31	31	31	31	31	31	31	31	31	31
C0035	1	1	1	1	1	1	1	1	1	1	1	1
C0074	0	1	0	0	0	0	0	0	0	0	0	0
C0225	1	1	1	1	1	1	1	1	1	1	1	1
G1203	1	1	1	1	1	1	1	1	1	1	1	1
G1218	1	1	1	1	1	1	1	1	1	1	1	1
G1295	16	16	16	16	12	13	16	16	16	16	12	12
G1297	12	12	12	12	8	12	14	14	14	14	10	10
G0286	1	1	1	1	1	1	1	1	1	1	1	1
G0472	1	1	1	1	1	1	1	1	1	1	1	1
G0473	2	2	2	2	2	2	2	2	2	2	2	2
G0475	1	1	1	1	1	1	1	1	1	1	1	1
G0476	0	0	0	0	0	0	0	0	0	0	0	0
G0595	1	1	1	1	1	1	1	1	1	1	1	1
G0596	1	1	1	1	1	1	1	1	1	1	1	1
G0597	1	1	1	1	1	1	1	1	1	1	1	1

Inventory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
G0598	1	1	1	1	1	1	1	1	1	1	1	1
G0599	2	2	2	2	2	2	2	2	2	2	2	2
G0732	1	1	1	1	1	1	1	1	1	1	1	1
G0733	1	1	1	1	1	1	1	1	1	1	1	1
G0734	4	4	4	4	4	4	4	4	4	4	4	4
G0835	1	1	1	1	1	1	1	1	1	1	1	1
G1095	2	2	2	2	2	2	2	2	2	2	2	2
G1878	1	1	1	1	1	1	1	1	1	1	1	1
G2002	9	9	9	9	9	9	9	9	9	9	9	9
G2004	9	9	9	9	9	9	9	9	9	9	9	9
G2343	1	1	1	1	1	1	1	1	1	1	1	1
G2415	1	1	1	1	1	1	1	1	1	1	1	1
G0288	5	5	5	11	5	5	5	5	5	5	5	10
G0023	1	1	1	1	1	1	1	1	1	1	1	1
G0062	1	1	1	1	1	1	1	1	1	1	1	1
G0109	3	3	3	3	3	3	3	3	3	3	3	3
G0110	2	2	2	2	2	2	2	2	2	2	2	2
G0181	3	3	3	3	3	3	3	3	3	3	3	3
G0212	1	1	1	1	1	1	1	1	1	1	1	1
G0213	1	1	1	1	1	1	1	1	1	1	1	1
G0164	127	127	127	127	127	127	127	127	127	127	127	127
G0171	145	145	145	145	145	145	145	145	145	145	145	145
G0765	2	2	2	2	2	2	2	2	2	2	2	2
G1043	45	38	31	39	36	33	27	27	27	27	24	24
G2197	108	108	36	36	36	36	36	36	36	36	36	36
G2427	1	1	1	1	1	1	1	1	1	1	1	1
G2428	1	1	1	1	1	1	1	1	1	1	1	1
G2517	2	2	2	2	2	2	2	2	2	2	2	2
G2978	1	1	1	1	1	1	1	1	1	1	1	1
G3063	6	6	6	6	6	6	6	6	6	6	6	6
G3131	7	7	7	7	7	7	7	7	7	7	7	7
G3135	1	1	1	1	1	1	1	1	1	1	1	1
G3142	4	4	4	4	4	4	4	4	4	4	4	4
G3775	2	2	2	2	2	2	2	2	2	2	2	2
G4248	1	1	1	1	1	1	1	1	1	1	1	1
G4391	11	11	11	11	11	11	11	11	11	11	11	11
G0082	2	2	2	2	2	2	2	2	2	4	4	4
G0089	2	2	2	2	2	2	2	2	2	2	2	2
G0100	21	21	21	18	11	11	8	21	18	16	14	14
G0073	1	1	1	1	1	1	0	0	0	0	0	0
G0094	0	0	0	0	0	0	2	2	2	0	0	0
G0097	1	1	1	1	1	1	1	1	1	1	1	1
G0099	1	1	1	1	1	1	1	1	1	1	1	1
G0111	2	2	2	2	2	2	2	2	2	2	2	2
G0096	1	1	1	1	1	1	1	1	1	1	1	1
G0097	1	1	1	1	1	1	1	1	1	1	1	1
G0089	22	22	15	22	15	15	15	15	15	15	15	15
G1376	35	35	29	20	25	25	25	25	25	25	25	25
G1762	10	16	16	10	10	10	10	10	10	10	10	10
G0373	0	0	0	0	0	0	0	0	0	0	0	0

BIOGRAFI PENULIS



Faldy Maulana Yuantoro lahir di Surabaya, pada tanggal 17 Juli 1996 sebagai anak ke-2 dari 3 bersaudara. Penulis menempuh pendidikan sekolah di SDN Pucang IV Sidoarjo (2002-2008), SMPN 1 Sidoarjo (2008-2011), dan SMAN 1 Sidoarjo (2011-2014). Setelah menyelesaikan pendidikan menengah, penulis melanjutkan pendidikan sarjana di Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya sejak tahun 2014 hingga 2018.

Selama menempuh pendidikan sarjana, penulis pernah melakukan kerja praktik (*On The Job Training*) di PT Indonesia Asahan Aluminium (Persero) pada Seksi *Spare Parts Warehouse* (SWH). Penulis juga aktif sebagai asisten Laboratorium *Quantitative Modelling & Industrial Policy Analysis* sejak tahun 2016 dan menjadi koordinator asisten pada tahun 2017. Selain aktif pada kegiatan akademik, penulis juga aktif dalam kegiatan organisasi di Himpunan Mahasiswa Teknik Industri ITS sebagai Ketua Pelaksana Lomba Keilmuan Teknik Industri se-Asia Tenggara (*Industrial Challenge* 2016) dan Wakil Kepala Divisi *Big Event IE Fair* (2016-2017). Penulis juga ikut serta dalam beberapa proyek pada bidang *Multi Criteria Decision Making*, *Metaheuristik*, *Data Analysis*, dan *Data Mining*, salah satunya pada penyelesaian permasalahan *human resource* di Pembangkit Jawa Bali Service (PJBS) PLTU Amurang, di Sulawesi Utara. Pada tahun 2017, penulis juga berhasil menjadi Juara Harapan II pada Kompetisi Rekayasa Kualitas Tingkat Nasional V tentang “Implementasi *Six Sigma* dan *Data Mining*” di Jakarta. Kritik dan saran penulis terima sebagai masukan untuk penelitian ini. Penulis dapat dihubungi melalui *e-mail* di faldymy@gmail.com.

